SOFT COMPUTING BASED MODELING OF DYNAMIC SYSTEMS

by

Mohammad Fazle Azeem

Electrical Engineering Department

Submitted

in fulfillment of the requirements of the degree of Doctor of Philosophy

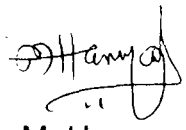
to the

Indian Institute of Technology, Delhi

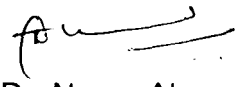
February 2000

Certificate

This is to certify that the thesis entitled *"Soft Computing Based Modeling of Dynamical Systems"* being submitted by **Mohammad Fazle Azeem** for the award of the degree of the **Doctor of Philosophy** to the Indian Institute of Technology, Delhi is a record of the bonafide research work he has carried out under our supervision. The results contained in this have not been submitted to any other University or Institute for the award of a degree or diploma.



(Dr. M. Hanmandlu)
Professor



(Dr. Nesar Ahmad)
Assistant Professor

Department of Electrical Engineering
Indian Institute of Technology, Delhi
New Delhi – 110 016

Acknowledgment

It is my great pleasure to express my appreciation to everyone who is directly or indirectly related with this thesis work. Many different people have made this thesis possible, none more so than my supervisors, Prof. M. Hanmandlu and Dr. Nesar Ahmad. Dr. Nesar Ahmad gave me firm understanding of soft computing and Prof. Hanmandlu always removed my mathematical hurdles and did hard work of proof reading of drafts. I am deeply grateful for all their support, encouragement and for being tolerant to my irregularities during the entire course of this work.

I would also like to express my thanks to SRC members, Prof. A.N. Jha, Prof. R. Balasubramanium and Dr. R.K.P. Bhatt, for their timely and useful suggestions.

I owe many thanks to Mr. R. S. Sanyal, Joint Director, Center for High Technology, Ministry of Petroleum and Natural Gas, Govt. of INDIA, for providing the operational data of FCCU.

I extend my thanks to my employer, Aligarh Muslim University (AMU), Aligarh, for sponsoring my Ph.D. program at IIT-Delhi.

I cannot forget to mention the name of Syed Atiqur Rahman, who is my friend, colleague and research mate, because I share all my worries, pleasures and even technical discussion with him, which provide me the matter to think.

I would like to express my thanks to my senior colleagues, Prof. Salman Beg, Dr. Anwaruddin Anwar and Prof. A.K. Gupta of AMU, Aligarh for their overwhelming support during the period of writing the thesis.

I want to express my sincere thanks to my wife who had given me a nice company throughout my uphill task to reach the peak of achievement and remained cool against all odds of my modish temperament. I would also like to express my thanks to my brothers and sisters, especially Mohammad Karme Azeem and Mohammad Taimoor Azeem, for their lovable support. I want to express my love to my two little daughters, Zainab and Nausheen, whose flowerish smiles and childish queries give me feeling of freshness.

Finally before closing this I cannot but think of my parents for what I am today.

Mohammad Fazle Azeem

Abstract

The emergence of Soft Computing paradigm as a consortium of Fuzzy logic, Neural Computing and Genetic algorithms has paved the way for handling complex dynamic systems. The present thesis makes an attempt to tackle some problems such as input selection, structure determination and model learning, associated with modeling of complex dynamic systems from the imprecise and uncertain operating data consisting of input-output pairs. The issues arising from these problems form the basis of this thesis.

As part of input selection, new criteria have been proposed based on the fuzzy curve approach for evaluating the significance of inputs. Fuzzy curve is the output Approximate Fuzzy Data Model (AFDM) and the changes in the output of AFDM over the range of input yields the ratios, which form the basis for input selection.

The structure determination is concerned with the choice of the number of rules for a particular model. In order to widen the choice, a Generalized Fuzzy Model (GFM), which may be categorized into four classes, encompassing Takagi-Sugeno (TS) Model and Compositional Rule of Inference (CRI) Model as special cases has been proposed. In order to learn the parameters of GFM, a Generalized Radial Basis Function (GRBF) Network has been suggested. It is shown that for any solution using GRBF, the error surface generated is symmetric. Finally, the equivalence of GRBF with GFM has an important bearing on the learning ability of Generalized Adaptive Neuro-Fuzzy Inference Systems (GANFIS).

For determining the structure, we have used clustering by Fuzzy C-means, Modified Mountain techniques in addition to Fuzzy curve approach for partitioning the input-output hyperspace into a number of clusters, which correspond to the number of rules. To provide a trade off between the computational complexity and model performance a Structure Identification Criterion (*SIC*) has been proposed.

The parameters of GFM are learned through the Least Square Estimation (LSE) and gradient Descent (GD) learning. In case, the models fail to achieve the targeted performance, the local learning algorithm is combined separately with Simulated Annealing (SA) to give SA hybrid and with Genetic Algorithm (GA) to give GA hybrid.

The modeling methodology has been tested on the practical plant like Fluidized Catalytic Cracking Unit (FCCU).

Dedicated to _____

My Parents

For their sacrifice, support and pain incurred by my absence during the Ph.D. program.

CONTENTS

List of Most Commonly Used Symbols and Abbreviations

Symbol/ Abbri.	Explanation
Chapter 1	
y_p	p^{th} output
l	Number of outputs
u_q	q^{th} input
r	Number of inputs
A_{ip}^k	Fuzzy set for i^{th} premise variable in k^{th} rule for p^{th} output
B_p^k	Consequent fuzzy set in k^{th} rule for p^{th} output
$\mu_{ip}^k(x_i)$	Membership function for fuzzy set A_{ip}^k
$\mu_p^k(\boldsymbol{x})$	Firing strength of k^{th} rule for p^{th} output
A_p^k	Fuzzy set corresponding to $\mu_p^k(\boldsymbol{x})$ in premise variable space
$f_p^k(\boldsymbol{x})$	Local model that represent consequent part of TS model in k^{th} rule for p^{th} output
CRI	Compositional Rule of Inference
TS model	Takagi-Sugeno model
Chapter 2	
n	Number of premise variable
n_k	Number of premise variable in k^{th} rule

m	Number of rules
M	Number of data set
X_i^k	Fuzzy set defined at x_{ik} along $y = y_k$ in x_i-y space
$\mu_{X_i^k}$	Membership function for fuzzy set X_i^k
$y_i^o(x_i)$	Output of AFDM
AFDM	Approximate Fuzzy Data model
p^k	Probability of k^{th} datum
p	Maximum number of partitioning for premise variables
J	Objective function [i.e., normalized mean square error of the model defined by eqn. (5.1)]

Chapter 3

A_i^k	Fuzzy set for i^{th} premise variable in k^{th} rule
$\mu_i^k(x_i)$	Membership function for fuzzy set A_i^k
$\mu^k(\boldsymbol{x}^k)$	Firing strength of k^{th} rule
B^k	Consequent fuzzy set for in k^{th} rule
$\phi^k(y)$	Membership function for fuzzy set B^k
b_k	Centroid of B^k
v_k	Area of B^k

GFM	Generalized Fuzzy Model
WGTF	Width of Gaussian type function
RBF	Radial Basis Function
GRBF	Geenralized RBF
GRBFN	GRBF Network
σ_k	WGTF
$\varphi^k(x)$	GRBF operating on vector x
c_k	Center vector of GRBF $\varphi^k(x)$
a_k	vector formed by inverse of WGTF for GRBF $\varphi^k(x)$
l_k	Power vector for GRBF $\varphi^k(x)$

Chapter 4

AIC	Akaike Information Criterion
SIC	Structure Identification Criterion
J_{cy}	Objective function for fuzzy C-means clustering defined by eqn. (4.7)
S	Cluster validity function (defined by 4.13 and 4.22)

Chapter 5

GD	Gradient Descent
LSE	Least Square Estimate
GA	Genetic Algorithms
SA	Simulated Annealing

Chapter 6

FCCU	Fluidized Catalytic Cracking Unit
RRSF	Reactor-Regenerator-Stripper-Fractionator
CCR	Catalyst Circulation Rate
FPH	Feed Pre-Heat
DCS	Distributed Control System

List of Figures

Fig. 6.45: Premise membership functions for class II GFM rules, of $M_{10}(t)$ variable.

Fig.6.46: Plots of (a) y (————, solid line) and y^o (- - - - - -,dashed line) (b) *model error* for class II GFM of $M_{10}(t)$.

Chapter 1

ISSUES IN SOFT COMPUTING

1-1 Emergence of Soft Computing

Science has evolved from trying to understand and predict the behavior of the universe and systems within it. Much of this owes to the development of suitable models, which agree with the observations. These models are in symbolic form, which the humans use, and in mathematical form found from physical laws. Most systems are causal, which can be categorized as either *static* where the output depends on the current inputs, or *dynamic* where the output not only depends on the current inputs, but also on the past inputs and outputs. Many systems also possess unobservable inputs, which cannot be measured, but affect the system's output. These inputs are known as disturbances and aggravate the modeling process.

Especially modern control theory has made rapid strides in areas where the system is well defined, but has failed to cope with practicalities of many industrial processes and systems, despite the development of a huge body of mathematical knowledge. There are undoubtedly many reasons for this, but fundamentally it is the lack of detailed structural knowledge of the processes and systems, inhibiting parametric/structural uncertainties and imprecision, which precludes the use of available

knowledge. Despite this, it is observed that in a fairly large class of industrial situations an operator may be able to control the process manually (or semi-automatically) based on his experience and/or knowledge of the plant. An operator's ability to interpret linguistic statements about the process and to reason in a qualitative fashion prompts the question " Can we make use of this information in an intelligent system?". It is envisaged that the operator is able to perform his/her task on the basis of learning and approximate reasoning capabilities of human brain that accounts for intelligence, which is not considered normally when a precise mathematical model is derived.

To cope up with the complexity of dynamic systems, there have been significant developments in modeling and control during the last two decades. Attempts are now being made to incorporate new paradigm -*Fuzzy Logic* (**FL**) and *Artificial Neural Networks* (**ANN**) to handle uncertainty and impreciseness of the real world systems (processes/plants). Fuzzy Logic tries to emulate the approximate reasoning ability of human brain. An Artificial Neural Network generally called Neural Network (**NN**) tries to simulate the learning ability of a human brain. It is envisaged that if both the approximate reasoning power of **FL** and learning ability of **NN** are combined together, it may be easier to handle complex systems. Recent trends aim at combining an evolutionary computing tool known as *Genetic Algorithms* (**GAs**) with **NN** and **FL** to optimize the model building as well as controller design for complex systems. These sets of tools/techniques are generally known as *Soft Computing* (**SC**). In essence, SC is a consortium of methodologies that provide a foundation for the conception, design, and deployment of intelligent systems. The principal partners in this consortium are fuzzy logic, neuro-computing, genetic algorithms and probabilistic reasoning. These

methodologies are for the most part complementary rather than competitive. Increasingly, the methodologies are used in combination, giving rise to what is referred to as a *hybrid system*. The most visible systems of this type are Neuro-Fuzzy or/and Fuzzy-Neuro systems **[Nauck'95; Gupta'98]**, that are mostly fuzzy rule-based systems employing neural network techniques for tuning and optimization. These new trends have yet to enforce their full impact on the system theory.

1-2 Role of Soft Computing in System Modeling

The pioneering work of Takagi and Sugeno **[Takagi'85]** in the fuzzy identification applied to modeling and control has led to several works reported in the literature **[Fukuda'92; Liska'94; Lotfi'96; Yager'92; Yoshikawa'94 Sugeno'93; Wang'95]**. This is a multi-model based approach **[Pedrycz'96].** The basic idea in this approach is to decompose the complicated input space into subspaces and then approximate the system in each subspace by a simple linear regression model called local model. The overall fuzzy model is considered as a combination of interconnected subsystems with simpler models.

Work of Jang **[Jang'93]** and Hunt **[Hunt'96]** has shown that with some minor restrictions, the functional behavior of fuzzy logic and Radial Basis Function (RBF) networks is actually equivalent. This functional equivalence of two models, stemming out of two different origins, is of great importance for providing learning ability to FLS **[Jang'93a; Nauck'94, 95, 97, 98]** and their equivalence needs further probe for more generalization.

Following Jang's Adaptive Network based Fuzzy Inference System (ANFIS) Chin et al., **[Chin'98]** have used it in multi target tracking in conjunction with Joint Probabilistic Data Association (JPDA) / Nearest Neighbour Probabilistic Data Association (NNPDA) algorithms. This application has its own limitation.

Chao et.al., **[Chao'96]** have used a fuzzy neural network system(**FNNS**) to implement fuzzy logic system. To reduce the complexity, they have used fuzzy similarity measures to eliminate the redundant fuzzy logic rules. Fuzzy similarity measures are also used to combine similar input linguistic fuzzy sets. With this they found that FNNS still has the desired performance with fewer fuzzy rules and adjustable parameters.

We now describe briefly the contributions made in the application of GA on some systems:

To optimize FLS for various applications, use of genetic algorithms (GAs) has been reported in the literature. Ishibuchi et al. **[Ishibuchi'95]** have applied these techniques for the classification problem. At US Bureau of mines, Karr and his Associates **[Karr'91a, 91b, 93]** have used GAs to design an adaptive fuzzy logic controller (FLC) for a laboratory acid-base experiment by simultaneously tuning the membership functions and the number of rules.

Machado and Rocha **[Machado'92]** have proposed a combination of FNN with GA **[Goldberg'89]**, producing a flexible and powerful learning paradigm, which they call Evolutive Learning. The evolutive learning combines complementary tools such as inductive learning (through the adjustment of synaptic weights) and deductive learning (through the modification of the network topology) to obtain an automatic adaptation of a system knowledge to the problem domain environment.

Perneel et.al., **[Perneel'95]** have developed a fuzzy logic decision making system using heuristic search algorithm. They tested and compared two optimization techniques, viz., GA and gradient descent algorithm to tune the parameters of the decision-making system for improving its performance.

1-3 Issues in Fuzzy Modeling

In **[Wang'92a]**, a table-lookup scheme is used to generate fuzzy rules directly from numerical examples and by the Stone-Weierstrass theorem and it is proved that a fuzzy inference system is a universal approximator. A heuristic method for generating Takagi-Sugeno (TS) fuzzy rules from numerical data is presented in **[Nozaki'97]**, and the consequent parts of TS fuzzy rules are then translated into linguistic representation. Hong and Lee **[Hong'96]** have pointed out that the drawback of most of the fuzzy inference systems is the need to predefine membership function and fuzzy rules from the numerical data in terms of linguistic expression and the need to make fuzzy reasoning. On the other hand, with a predefined membership function the number of fuzzy rules increases exponentially with the increase in input variables. Though rules do not cover rectangular shapes in the input-output hyperspace, the optimum number of rules is arranged at appropriate positions in the fuzzy space **[Azeem'98a]**.

To optimize an adaptive fuzzy system used in modeling and control, it is possible to adjust the following parameters in order to achieve the desired system performance:

- the shape of membership function;
- the number of rules used (structure);
- the inference mechanism;

The effect of changing the membership functions is predominant over the remaining two parameters, but size of rule base affects the computational time. For real time applications, optimization of the first two parameters, namely, the membership function and the number of rules is required for any fuzzy reasoning methods (inferencing mechanism) **[Mizumoto'88]**. During the last decade, researchers have been trying to integrate FLS with neural networks, simulated annealing and genetic algorithms (GAs). These efforts are attributed to the real time application for the adaptive behavior of FLS.

1-4 Dynamic Fuzzy Model

Linear / nonlinear dynamic system models may be represented by mapping from the input space to the output space, which we call as function approximation. Systematic analytical work on function approximation of fuzzy systems **[Zeng'94, 95a, 95b, 96a, 96b, 96c]** has shown that fuzzy systems under certain fuzzy reasoning **[Mizumoto'88; Nauck'92]** turn out to be the universal function approximators.

To construct fuzzy rules for a Multi-Input and Multi-Output (MIMO) system which can be split up into several Multi-Input and Single-Output (MISO) systems, consider a Non-linear Auto-Regressive Moving Average (NARMA) model **[Park'95]** representing a MIMO system:

$$y_p(t) = f_p\begin{pmatrix} u_1(t),..,u_1(t-\tau_{i1}),u_2(t),..,u_r(t-\tau_{ir}),y_1(t-1) \\ ,..,y_1(t-\tau_{o1}),y_2(t-1),..,y_l(t-\tau_{ol}) \end{pmatrix} \quad p=1,2,...,l \quad ...(1.1)$$

where $u_q;(q=1,\ldots,r)$ and $y_p;(p=1,\ldots,l)$ denote the inputs and outputs respectively. τ_{iq} and τ_{op} are the corresponding delays.

The above system can be represented either by:

Compositional Rule of Inference (CRI) Model

$$R_p^k : \text{if } x_1 \text{ is } A_{1p}^k \wedge x_2 \text{ is } A_{2p}^k \wedge \ldots \wedge x_n \text{ is } A_{np}^k \quad \text{then } y_p \text{ is } B_p^k \qquad \ldots(1.2)$$

or by *Takagi-Sugeno (TS) Model*

$$R_p^k : \text{if } x_1 \text{ is } A_{1p}^k \wedge x_2 \text{ is } A_{2p}^k \wedge \ldots \wedge x_n \text{ is } A_{np}^k \quad \text{then } y_p \text{ is } f_p^k(\boldsymbol{x}) \qquad \ldots(1.3)$$

where, A_{ip}^k are linguistic labels of fuzzy sets describing the qualitative state of the input variable x_i for p^{th} output variable in k^{th} rule R_p^k, $\wedge$ is a fuzzy conjunction operator (usually of T-norm). The firing strength of the k^{th} rule is obtained by taking the T-norm (usually min or product operator) of the membership functions of the premise parts of the rule as:

$$\mu_p^k(\boldsymbol{x}) = \mu_{1p}^k(x_1) \wedge \mu_{2p}^k(x_2) \wedge \ldots \wedge \mu_{np}^k(x_n) \qquad \ldots(1.4)$$

where, $\mu_{ip}^k(x_i)$ is the membership function of fuzzy set A_{ip}^k. The firing strength of k^{th} rule of p^{th} output variable y_p is also represented as a fuzzy set $A_p^k \subset R^n$ in the input space and each rule is premised on the input vector $\boldsymbol{x}$. Hence eqn.(1.2) and eqn.(1.3) can be rewritten as:

CRI-Model

$$R_p^k : \text{if } \boldsymbol{x} \text{ is } A_p^k \text{ then } y_p \text{ is } B_p^k \qquad \ldots(1.5)$$

or *TS-Model*

$$R_p^k : \text{if } \boldsymbol{x} \text{ is } A_p^k \text{ then } y_p \text{ is } f_p^k(\boldsymbol{x}) \qquad \ldots(1.6)$$

Each rule maps fuzzy subsets in the input space $A_p^k \subset R^n$ to a fuzzy subset in the output space $B_p^k \subset R$ for *CRI-Model* or a local model $f_p^k(x)$ for *TS-Model*, where $k = 1, \ldots, m$, m being the number of rules. A linear form of $f_p^k(x)$ is as follows:

$$f_p^k(x) = b_{0p}^k + b_{1p}^k x_1 + \ldots + b_{np}^k x_n \qquad \ldots(1.7)$$

The premises of the rules, which represent delays as well as the order of dynamic systems, for the NARMA model of a complex system are denoted by:

$$\begin{aligned} x &= \{x_1,\ldots,x_{(\tau_{i1}+\ldots+\tau_{ir}+r)}, x_{(\tau_{i1}+\ldots+\tau_{ir}+r+1)},\ldots,x_n\} \\ &= \{u_1(t),\ldots,u_1(t-\tau_{i1}), u_2(t),\ldots,u_r(t-\tau_{ir}), y_1(t-1),\ldots,y_1(t-\tau_{o1}), y_2(t-1),\ldots,y_l(t-\tau_{ol})\} \end{aligned}$$

where, $$n = \tau_{i1} + \ldots + \tau_{ir} + \tau_{o1} + \ldots + \tau_{ol} + r \qquad \ldots(1.8)$$

1-5 Motivation

Since most of the practical systems are ill defined due to parametric and/or environmental uncertainty, researchers have shown that Soft-computing techniques are well suited to handle these uncertainties.

If *apriori* information about the system is not available, only the input-output data sets are available with time, one of the problems we face in modeling such systems is the lack of information regarding the number of inputs to a model. In an attempt to solve this problem we evaluate the significance of the past inputs, the past outputs and the current inputs with respect to current outputs.

Adaptive Neuro-Fuzzy Inference System (ANFIS) of Jang **[Jang'93a],** which uses Gaussian membership function for the premise variables, poses a constraint on the FIS. It has been shown in the literature that a wide variety of membership functions can

be used appropriate to the application. A general trend is that the initial fuzzy partitioning of Universe of Discourse (UoD) carried out arbitrarily with equal space and the same shape, may end up in local minima during adaptation. The number of rules is taken arbitrarily. The consequent part of each rule is taken as a linear function of inputs based on TS-model.

An attempt has been made in this thesis to overcome all these discrepancies,. The following section outlines the issues taken up in the thesis and provides the directions to solve these issues.

1-6 Issues of the thesis

This thesis makes an attempt to improve the modeling capabilities of *Fuzzy Inference System* (FIS). In practice, dynamic system modeling is based on some *apriori* knowledge and input-output data of the system. Here we assume that *apriori* knowledge is not available but only the input-output data sets of the system are available. The issues and directions of the thesis are as follows:

1. The methods of Takagi-Sugeno **[Takagi'85]** and Sugeno-Yashikawa **[Sugeno'93]** for input selection requires the maximum of $n\{(2p-1).n+1\}/2$ and $n(n+1)/2$ fuzzy systems respectively to be tested for n input variable identification and 2^p maximum partitions.. For a system with a large number of input candidates, building such a large number of fuzzy models is not practical. The method of Lin and Cunningham **[Lin'95]** lists only a comparative significance of each input candidate without any further procedure for input selection. To solve this problem, the concept of fuzzy curve is applied on dynamic systems to evaluate the significance of input candidates before modeling the system. In this thesis new criteria are proposed for input variable

selection on the basis of significance evaluated from fuzzy curves. This approach is computationally simple and the time complexity is linear with respect to the number of input candidates.

2. There are plenty of choices of membership functions varying from trigonometric to trapezoidal. Most of them are not in closed form suitable for differentiation, which is the requisite during the gradient descent learning. It is well known fact in the literature that selection of membership function is application dependent. So a generalized form of membership function called Gaussian type function is proposed which caters to a wide range of membership functions from triangular to trapezoidal.
3. A Generalized Fuzzy Model (GFM) is proposed to inherit the complementary properties of TS Model as well as CRI model. For learning GFM need arises for a Generalized Adaptive Neuro-Fuzzy Inference System (GANFIS). Implementation of GANFIS presupposes the architecture in the form of the Generalized Radial Basis Function Network (GRBFN).
4. Since there is no generalized approach for the determination of an optimal rule set, some clustering is required to determine the optimum number of rules from the central positions in the input-output hyperspace, based on the optimization of certain objective function.
5. While implementing GANFIS, we may end up in local minima. The local learning algorithms are hybridized with some global learning algorithms in order to overcome the problem of local minima and to achieve better minima in finite time.

1-7 Organization of the Thesis

It may be mentioned that the results of earlier chapters require the methods proposed in the later chapters and vice versa. As a result, the breakup of the thesis into various chapters is only for the convenience of presentation, but does not reflect the hierarchical dependence of the chapters. The chapter wise breakup of the thesis is as follows:

Chapter 2

This chapter is mainly concerned with input selection. The complexity of input selection methods of Takagi-Sugeno **[Takagi'85]** and Sugeno-Yasukawa **[Sugeno'93]** is analyzed. The concept of Approximate Fuzzy Data Model (AFDM) is introduced in this chapter for generating the fuzzy curve and new criteria are proposed for input selection by defining a ratio of change in AFDM output over the range of the input candidate.

Chapter 3

In this chapter, a Generalized Fuzzy Model (GFM) is proposed using the consequent regression of Takagi Sugeno (TS) Model and area of consequent membership function of Compositional Rule of Inference (CRI) model.

To learn GFM adaptively, a Generalized Radial Basis Function (GRBF) network is proposed. This chapter investigates the issue of normalization vs. non-normalization in the context of RBF networks.

Chapter 4

In order to obtain an optimal number of fuzzy rules, in terms of model performance and computational complexity, Structure Identification Criterion (*SIC*) is defined, for getting an optimal number of clusters. The Akaike's Information Criterion (*AIC*) and cluster validity are compared with *SIC*. Two types of clustering techniques, namely,

Fuzzy C-means clustering and Modified Mountain clustering are used. A comparative study of different methodologies, namely, fuzzy-curve, Fuzzy C-means clustering and Modified Mountain clustering with regard to the initialization of GRBF network is also made in this chapter.

Chapter 5

In this chapter, a hybrid of two basic learning algorithms, namely Gradient Descent (GD) and Least Square Estimation (LSE), is used to learn GFM. Next, this hybrid is combined with either Genetic Algorithm (GA) to yield GA hybrid or combined with Simulated Annealing (SA) to yield SA hybrid for the purpose of achieving the targeted performance of the model.

Chapter 6

In this chapter, a case study of Fluidized Catalytic Cracking Unit (FCCU) of a petroleum refinery is presented. The objective of this study is to evaluate the proposed modeling approach on a practical plant.

Chapter 7

Conclusions of this thesis along with the suggestions for future work form this chapter.

Chapter 2

INPUT VARIABLE SELECTION

2-1 Introduction

To build a fuzzy model, the first step is to select significant inputs among many input candidates **[Takagi'85, Xu'87, Sugeno'93, and Lin '95]**. In the context of building a dynamic fuzzy model for a time delay system, the past system inputs are the input candidates for selection. Significant past system inputs are attributed to time delays in the system model. On the other hand, to incorporate the dynamics of a system in the model, the past system outputs become the input candidates for the system model since the significant past system outputs contribute to the dynamics of a system. This leads to linear/nonlinear Auto-Regressive Moving Average (ARMA) model **[Park'95].** We can identify the significant inputs to the model by using any identification algorithm **[Takagi'85, Xu'87, Sugeno'93]**. But, all these algorithms either fail to select the exact number of inputs **[Xu'87]** to the model or involve expensive computations **[Takagi'85, Sugeno'93]**. The models of Xu and Lu **[Xu'87]** are based on two input variables with complete rule base, and are therefore not suitable for multivariable systems. Takagi and Sugeno **[Takagi'85]** have proposed a method that requires a maximum of $n\{(2p-1)n+1\}/2$ test models for n input candidates and a maximum of 2^p partitions.

Sugeno and Yasukawa **[Sugeno'93]** have proposed a method, for input selection, decoupling the number of partitions. This method requires $n(n+1)/2$ test models. The method of Lin and Cunningham **[Lin'95]** lists only a comparative significance of each input candidate without any further test for input selection. In this chapter we propose new criteria for input variable selection on the basis of significance evaluated from fuzzy curves. As will be shown this approach is computationally simple and the time complexity is linear with respect to the number of input candidates.

For any model there are, of course, a number of possible input candidates, which should be restricted to a certain number. We select a set of input variables, which affects the output, among the given possible input candidates. In a conventional black box approach in system theory, the models are based on pre-assigned input-output variables. Generally speaking, some criterion is required to evaluate the performance of a model. Output error, i.e., the difference between model output and real output may be used in the criterion. However, as far as input variable selection is concerned, it is well known that only output error cannot be used. So, a special criterion needs to be devised.

Lin and CunninghamIII **[Lin'95]** have proposed a fuzzy curve approach for input variable selection. They have tested its efficacy on static systems. They have arbitrarily chosen the inputs among the tested input candidates according to the order of their significance based on fuzzy curves. Their procedure lacks the steps for the exact selection of inputs to the model. It may be noted that the model error will be minimum, if the identified variables are the ones, which actually affect the output of the system. Moreover, if some variables are missing or some additional variables are identified, the model error will not be minimum in fuzzy modeling. We wish to propose criteria for

identification of only those variables, which actually affect the output of the system, among the input candidates using fuzzy curve. Fuzzy curves are obtained from the Approximate Fuzzy Data Model (AFDM) over the domain of input. Approximation comes from the assumption that output is evaluated for each input individually. We can interpret a fuzzy curve as unconditional expected value in the next Section.

The organization of this chapter is as follows: Section 2-2 introduces the Approximate Fuzzy Model (AFDM) to evaluate the relative significance of inputs. The output of AFDM yields the fuzzy curve. Two methods for input selection are discussed before presenting the proposed criteria for input selection using the fuzzy curve approach in Section 2-3. A description of a few Examples of dynamic systems is provided in Section 2-4 for use in subsequent chapters. Simulation results of application of input identification to these systems are given in Section 2-5. Finally, conclusions are drawn in Section 2-6.

2-2 Approximate Fuzzy Data Model (AFDM)

AFDM is used to evaluate the relative significance of input candidates x_i $(i = 1, \ldots, n)$. For this purpose, consider a multi-input and single-output (MISO) system. For each input x_i, there is one rule in x_i - y space for each set of data points $(x_{ik}, y_k)(k = 1, 2, \ldots, M)$, of the following form:

$$R_i^k : \text{IF } x_i \text{ is } X_i^k \text{ THEN } y \text{ is } y_k, \qquad (k = 1, 2, \ldots, M.) \qquad \ldots(2.1)$$

where X_i^k is the fuzzy set representing the membership function, for input variable x_i, $\mu_{X_i^k}$ corresponding to data point x_{ik} and a Gaussian form is defined by:

$$\mu_{X_i^k}(x_i) = \exp\left(-\left(\frac{x_{ik} - x_i}{b}\right)^2\right) \quad k = 1, 2, \ldots, M. \qquad \ldots(2.2)$$

In general, $\mu_{X_i^k}(x_i)$ may be made to take any shape including triangular, trapezoidal, etc., by changing the power of Gaussian function [See the generalized Gaussian function given in eqns (3.21) and (3.22)]. In eqn.(2.2) a value of b is taken in the range 10% to 20% of the length of input interval of x_i, for $x_i = x_{ik}$, $\mu_{X_i^k}(x_{ik}) = 1.0$. The defuzzified output for any value of x_i over the range can be obtained by the weighted sum method.:

$$y_i^o(x_i) = \sum_{k=1}^{M} \frac{\mu_{X_i^k}(x_i)}{\sum_{k=1}^{M} \mu_{X_i^k}(x_i)} \cdot y_k \qquad \ldots(2.3)$$

With the above procedure an output is obtained, which can be plotted over the range of x_i to yield a curve called "fuzzy curve". We note here that fuzzy curve: is the output of AFDM, comprising rules corresponding to each data point in x_i - y space In AFDM, .the input influence is treated as a Gaussian function given by eqn. (2.2)

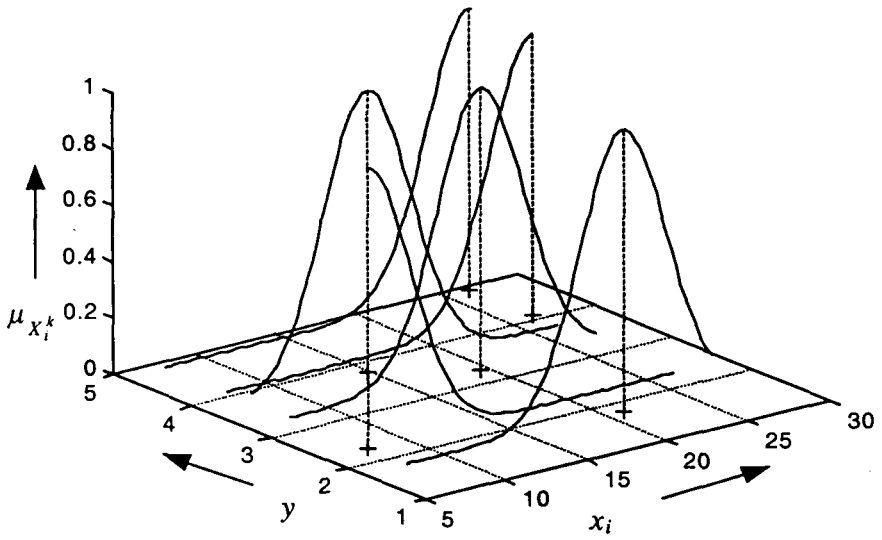


Fig.2.1. Premise (for the partial data set {x_1-y} from Table 2-1) membership function on the x_i-y plane.

TABLE 2-1

S.No.	x_1	x_2	x_3	y	S.No.	x_1	x_2	x_3	y
1.	**14.9**	2.0	9.3	**3.8**	11.	6.9	2.3	10.3	3.3
2.	16.6	1.7	5.8	3.7	12.	7.4	1.9	9.4	2.8
3.	21.3	2.1	9.1	3.0	13.	11.3	2.0	8.4	3.0
4.	24.3	2.9	7.4	4.7	14.	17.6	2.0	8.2	2.7
5.	**26.6**	1.7	4.8	**4.1**	15.	**19.5**	2.4	6.5	**3.3**
6.	23.2	1.3	4.7	3.1	16.	21.6	2.5	5.1	4.9
7.	22.2	2.5	4.5	3.1	17.	**26.5**	1.9	6.3	**4.8**
8.	18.1	2.5	5.9	2.5	18.	26.4	2.2	8.1	4.4
9.	13.7	2.8	7.9	3.3	19.	23.1	3.8	4.9	2.2
10.	**7.7**	2.6	9.3	**2.3**	20.	**21.1**	2.5	5.1	**1.8**

Consider a multi-input and single-output (MISO) system. Table 2-1 shows input-output data with n=3, M=20. For each input x_i (i=1, 2, 3), we plot the m data points in x_i-y space. Figure 2.1 shows the premise membership function for the partial data set $\{x_1\text{-}y\}$ on the x_1-y plane. Figure 2.2a shows the plots of data points in x_1-y, x_2-y, and x_3-y spaces. These plots have different scales. We will take care of these scales by normalizing in x and y dimensions, while devising the criteria for input selection.

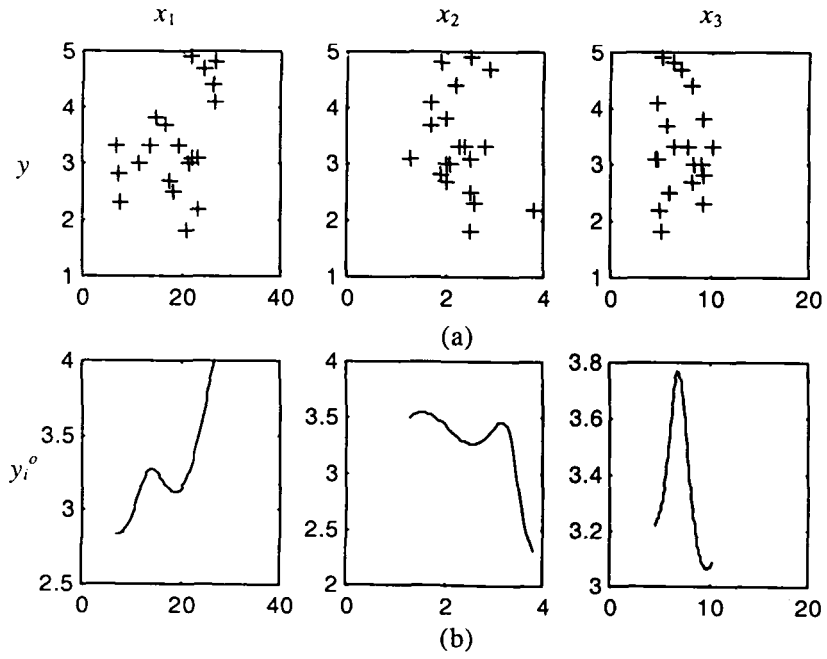


Fig. 2.2 : (a) Data points plotted in x_1-y, x_2-y, and x_3-y space (b) fuzzy curves y_1^o, y_2^o, and y_3^o

Figure 2.2(b) shows fuzzy curves y_1^o, y_2^o, and y_3^o for the data in Table 2-1. If the fuzzy curve for a given input candidate is flat, then this input candidate has little influence on the output and it is not a significant input. If the range of fuzzy curve y_i^o is near about the range of output data y, the corresponding input data x_i can be considered most important to the output variable. Fuzzy curve depicts the approximate change of output over the range of input x_i. Importance of input candidate x_i is decided according to the range covered by its fuzzy curve y_i^o.

2-2.1 Fuzzy Curve as Conditional Expected Value

As we know a fuzzy set is concerned with the degree of association of a property and probability deals with the occurrence of that property **[Kosko'97].** Further, each rule is a fuzzy subset and all rules in a domain constitute a fuzzy system or fuzzy model. All fuzzy subsystems have occurrences with some probability and are, therefore, of probabilistic in nature. In view of this, fuzzy system computes the expectation, which is an estimation or function approximation. Consequently, fuzzy curve being an approximation is the result of expectation as proved below.

Theorem 2.1: *In x_i - y space with M data, we define the conditional probability for k^{th} datum at any point in x_i - dimension by* (see Fig. 2.1)

$$p_{x_i}^k\left(x_i \middle| y = y_k\right) = \frac{1}{M}\mu_{X_i^k}(x_i) = \frac{1}{M}\exp\left(-\left(\frac{x_{ik} - x_i}{b}\right)^2\right) \qquad ...(2.4)$$

and the unconditional probability for k^{th} datum at any point in x_i - dimension by

$$p_{x_i}^k(x_i) = \sum_{k=1}^{M} p_{x_i}^k(x_i|y = y_k) = \frac{1}{M}\sum_{k=1}^{M} \mu_{X_i^k}(x_i) \qquad \text{...(2.5)}$$

Similarly, the unconditional probability for k^{th} datum at any point in y-dimension is defined by

$$\left.\begin{aligned} p_y^k(y = y_k) &= 1 \\ p_y^k(y \neq y_k) &= 0 \end{aligned}\right] \qquad \text{...(2.6)}$$

Then the output of AFDM $y_i^o(x_i)$ described by eqn.(2.3), which maps $x_i \in \mathcal{R}$ to $y_i^o \in \mathcal{R}$ with mapping F given by $F : x_i \rightarrow y_i^o$, is the conditional expected value of y, i.e., $E[y|x_i]$.

Proof: In discrete form, the conditional expected value of y is written as follows

$$E[y|x_i] = \sum_{k=1}^{M} y_k\, p_y^k(y = y_k|x_i) \qquad \text{...(2.7)}$$

Applying the Bayes' theorem with joint statistics, we obtain

$$E[y|x_i] = \sum_{k=1}^{M} y_k \frac{p_{x_i}^k(x_i|y = y_k)p_y^k(y = y_k)}{p_{x_i}^k(x_i)} \qquad \text{...(2.8)}$$

In view of the definitions [eqn. (2.4) to eqn. (2.6)], eqn (2.8) can be written as:

$$\begin{aligned} E[y|x_i] &= \sum_{k=1}^{M} y_k \frac{\frac{1}{M}\mu_{X_i^k}(x_i).1}{\frac{1}{M}\sum_{k=1}^{M}\mu_{X_i^k}(x_i)} \\ &= \sum_{k=1}^{M} \frac{\mu_{X_i^k}(x_i)}{\sum_{k=1}^{M}\mu_{X_i^k}(x_i)} y_k \end{aligned}$$

$$E[y|x_i] = y_i^o(x_i) \qquad \text{...(2.9)}$$

This proves that the conditional expected value of y is a particular point on the fuzzy curve corresponding to the input x_i.

2-3 Input Selection Criteria

In this Section we first investigate the number of the test models required for input selection by Takagi-Sugeno method **[Takagi'85]** and Sugeno-Yasukawa method **[Sugeno'93]** existing in the literature. Later we present the criteria for input selection using the information obtained from fuzzy curve and investigate the number of test models with the proposed method of input selection.

We will make a comparison of computational complexity associated with the above methods.

2-3.1 Takagi-Sugeno Method

Takagi and Sugeno **[Takagi'85]** have proposed a method for input selection based on heuristic search, which selects the input giving the best model performance. Here, we use the notation **Model *s-l*** to denote the test model at s^{th} stage obtained by adding l^{th} input candidate to the optimal model of $(s\text{-}1)^{th}$ stage. In stage 1, n models (Model 1-l; l= 1 to n) are built for n input candidates with 2 fuzzy partitions for each input candidate. Out of these models, the one with the minimum performance is selected as an optimal model for the next stage. Let the i^{th} input candidate be selected corresponding to the optimal Model 1-i. In stage 2 again n models (Model 2-l) are built with 2 fuzzy partitions of selected i^{th} variable in the previous stage and 2 fuzzy partitions for the rest of the input

candidates. For Model 2-i the number of partitions for i^{th} variable is 4 (i.e., 2^2). Again, in stage 2 the model with the best performance is selected. Let the i^{th} variable be selected again for the best performance of Model 2-i. In stage 3 again n models (Model 3-l) are built with 4 fuzzy partitions of i^{th} variable and 2 fuzzy partitions for the rest of the input candidates. For the Model 3-i the number of partitions for i^{th} variable is 8 (i.e., 2^3). In stage 3 the model with the best performance is selected. Let the j^{th} variable be selected corresponding to the best performance of Models 3-l. The input selection procedure in different stages is shown in Fig. 2.3, where the solid arrows show the path of selection. In each stage either a new variable is selected or the partitions get doubled for any of the selected inputs in the previous stages.

If the maximum number of partitions is 2^p and the maximum number of stages in which a variable can be selected is p, then the maximum number of stages is np. Until $n(p\text{-}1)$ stages, the maximum number of models to be tested is $n^2(p\text{-}1)$. In $n(p\text{-}1)+1$, $n(p\text{-}1)+2$, . . . , np stages, the number of models to be tested are n, $(n\text{-}1)$, . . . ,1 respectively.

So, the maximum number of models to be tested $= n^2(p\text{-}1) + [\, n + (n\text{-}1) + \ldots + 1 \,]$

$= n^2(p\text{-}1) + [\, n(n+1)/2 \,]$

$= n \{ (2p\text{-}1)n + 1 \} / 2$

Hence, the number of test models is of the order $\mathcal{O}\left(n^2 p\right)$.

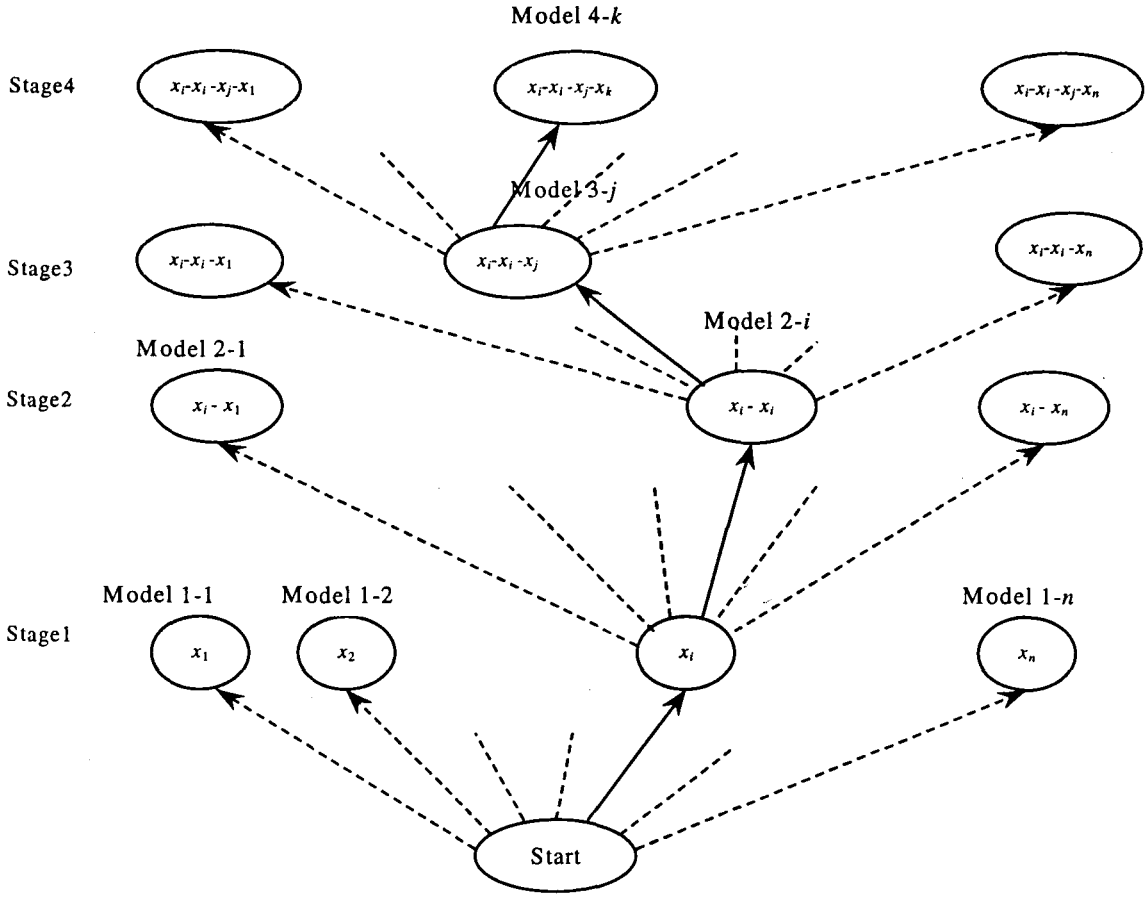


Fig. 2.3 : TS method of input selection

2-3.2 Sugeno-Yasukawa Method

Sugeno and Yasukawa **[Sugeno'93]** have considered an arbitrary number of fuzzy partitions for input selection and used the same heuristic search method. In each stage a new input is selected. For n input candidates there are n stages. In stage 1, the number of test models is n. In stage 2, the number of test models is n-1. In a similar way the number of test models in stage n, is 1.

So, the total number of models to be tested $= n + (n\text{-}1) + \ldots + 1$

$= n(n+1) / 2$.

Hence, the number of test models is of the order $\mathcal{O}(n^2)$.

The complexity of the above methods is of the order $\mathcal{O}(n^2 p)$ and $\mathcal{O}(n^2)$. Now we will present a method whose complexity is of the order $\mathcal{O}(n)$.

2-3.3 Fuzzy Curve Method

From the shape of the fuzzy curve the following inferences can be drawn:

i) Effect of change in input in the form of positive slope results in the increase of output and the negative slope results in the decrease of output with the increase in x_i.

ii) Based on the heuristic that fuzzy model will interpolate between maxima and minima, the minimum number of rules required to approximate each fuzzy curve can be determined from the number of maxima and minima on each fuzzy curve.

The inference (i) and (ii) though provide information, but they are not suitable for input selection. The reason is that curves may have the same number of maxima and minima and may have same absolute gradients.

iii) The importance of variable x_i, among the input candidates is evaluated from the approximate changes in the output of AFDM, i.e., c_i over the range of input x_i. c_i is defined as:

$$c_i = \frac{\mathbf{max}\left(y_i^o(x_i)\right) - \mathbf{min}\left(y_i^o(x_i)\right)}{\mathbf{max}\left(y_k\right) - \mathbf{min}\left(y_k\right)} \times 100 \qquad \text{...(2.10)}$$

The ranges of fuzzy curves in Fig. 2.2(b) are 1.15 for y_1^o, 1.23 for y_2^o, and 0.703 for y_3^o. Hence x_2, is the most significant, followed by x_1 and x_3 in that order. Here the following question arises:

1. Upto what significance can one select inputs to the model?
2. Whether the input candidate with higher significance (i.e., larger value of c_i) gives better model performance?

This can be judged by the following criteria proposed by us. Before applying the criteria, sort out all the input candidates for the model in descending order of their c_i.

Criterion1: Select all those input candidates for which $c_i \geq h$. where $h \in [0,100]$.

Value for "h" is application dependent. It also depends upon the number of input candidates as well as values of c_i's. Higher values of "h" will lead to less number of selected input candidates and vice-versa.

Criterion2: Among the selected input candidates, start modeling with the input candidate of highest value of c_i and go on adding input candidates one by one with the descending value of c_i. In doing so, accept only those input candidates which significantly improve the performance of model and reject those input candidates, inclusion of which deteriorate / do not improve the performance of model. Terminate the input selection when model performance reaches the target value.

With the *criterion1* if n variables are selected, we need to evaluate n test models with single input to find out the variable resulting in the minimum performance of model. On the basis of *criterion2* we need to evaluate a maximum of n-1 test models. With the proposed method the maximum number of test models to be evaluated is $2n$-1. Therefore, the maximum number of test model is linear with the number of input candidates, i.e., $\mathcal{O}(n)$.

The validation of the above criteria on few dynamical systems is given in the Section 2-5

2-4 Description of Some Dynamic Systems

Four different classes of dynamic systems are described in the following examples for validation of the proposed criteria.

Example 2.1:

We consider a two-input, and single output bilinear dynamical model **[Xu'87]** given below:

$$y(k) = 0.8y(k-1)u_1(k) + 0.5u_1(k-1)y(k-2) + u_2(k-4) + \alpha.e(k) \quad ...(2.11)$$

to provide the input-output data sequence, which is expressed as follows:

data 1 ($\alpha = 0$): { $y(k)$, $u_1(k)$, $u_2(k)$, $k = 1, 400$ } ...(2.12)

data 2 ($\alpha = 1$): { $y(k)$, $u_1(k)$, $u_2(k)$, $k = 1, 400$ } ...(2.13)

In the model (2.11), $e(t)$ is an uncorrelated random noise uniformly distributed over the interval (-0.08, 0.08), one can use the Gaussian noise as well. Therefore, data 1 are noise free and data 2 are noisy. Input $u_1(k)$, and $u_2(k)$ are both uncorrelated random sequences uniformly distributed over the interval (0.1, 0.9).

We consider the variables $u_1(t)$, . . ., $u_1(t\text{-}4)$, $u_2(t)$, . . ., $u_2(t\text{-}6)$, $y(t\text{-}1)$, . . ., $y(t\text{-}3)$ as input candidates to the model.

Example 2.2: Gas furnace

A benchmark problem of system identification given in **[Box'70]** is considered. The process in this example is of a gas furnace with single input $u(t)$, i.e., gas flow rate and single output $y(t)$, i.e., CO_2 concentration.

Since the process is dynamic, we consider as input candidates the eleven variables $u(t)$ to $u(t\text{-}6)$, and $y(t\text{-}1)$ to $y(t\text{-}4)$, to affect the present output $y(t)$.

Example 2.3: Human operation at a chemical plant

Here, we deal with a model of an operator's control action at the startup of a chemical plant **[Sugeno'93]**, which is meant for producing a polymer by the

polymerization of some monomers. Since the start up of the plant is very complicated, a man has to make the manual operations in the plant.

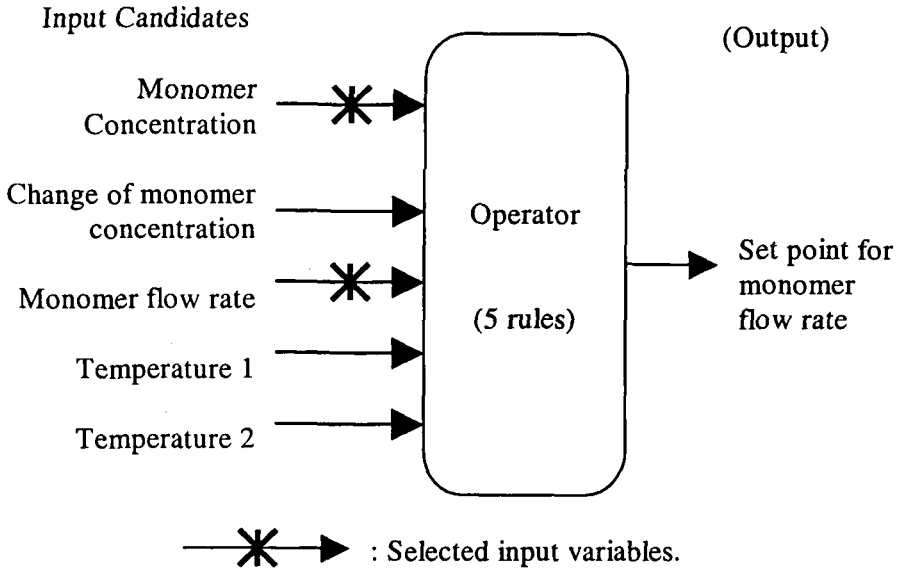


Fig. 2.4: Control action of an operator

As shown in Fig. 2.4 there are five input candidates which the human operator might refer to for control, and one output, i.e., his control action given as under:

u_1 : monomer concentration.

u_2 : change in monomer concentration.

u_3 : monomer flow rate.

u_4, u_5 : local temperatures inside the plant.

y : set point for monomer flow rate.

The operator determines the set point for the monomer flow rate and the actual value of the monomer flow rate for the plant controlled by PID controller.

Example 2.4: Daily Stock price

Finally, we take up the trend data of stock prices. Here, we use the daily data of a stock market. There are 100 data points taken from **[Sugeno'93]**. The data consist of ten inputs and one output. These are:

x_1 : The past change of moving average over a middle period;
x_2 : The present change of moving average over a middle period;
x_3 : The past separation ratio with respect to moving average over a middle period;
x_4 : The present separation ratio with respect to moving average over a middle period;
x_5 : The present change of moving average over a short period;
x_6 : The past change of price, for instance, change in one day before;
x_7 : The present change of price ;
x_8 : The past separation ratio with respect to moving average over a short period;
x_9 : The present change of moving average over a short period;
x_{10} : The present separation ratio with respect to moving average over a short period;
y : Prediction of stock price;

where the separation ratio is a value concerning the difference between a moving average of a stock price and the price of stock.

2-5 Simulation Results

A normalized mean square error J, is considered for the evaluation of the test models. Since the Class II Generalized Fuzzy model (GFM), gives the best performance, the results presented for input selection correspond to the Class II GFM, (See chapter 3 for details). The Initial parameters for all the test models are found by the fuzzy curve method. Further these initial rules are fine tuned using GD and LSE techniques (See chapter 5). The evaluation of the test model is based on normalized mean square error, and for stopping the input selection, the target value of J is set to 2×10^{-4}. Since we have

considered a limited number of input candidates in all the examples, we directly apply the *criteron2*.

Example 2.1

Fuzzy curve for all the input candidates is evaluated. Using the fuzzy curve the values of c_i for data 1 ($\alpha = 0$) and data 2 ($\alpha = 1$) are obtained from eqn.(2.10). For both the data we set the number of fuzzy rules for the model, from the maxima and minima of fuzzy curve (See Section 4-3.1), m=3. In Table 2-2, the values of c_i for each input candidate are given for both the data. The orders of all the input candidates according to the descending values of c_i are also given as shown in Table 2-2 and labeled as "Descending Order of c_I".

First, 80% of data is used for model learning, the rest 20% is used for model validation. The model performance, shown by + sign, with the each individual input candidate is plotted in Fig.2.5 and Fig.2.6 for data 1 (i.e., noise free data) and data 2 (i.e., noisy data) respectively. The solid line in Figs: 2.5 and 2.6 shows the performance of models with the application of *criterion2*. A circle on the solid line shows the acceptance of that input candidate as the input variable. In this example the model performance, on both the data, reaches its target value by adding the input candidates without any rejection and the input selection procedure is terminated after adding the variable $u_1(t\text{-}1)$. The selected variables for both the data are $y(t\text{-}1)$, $y(t\text{-}2)$, $u_2(t\text{-}4)$, $u_1(t)$ and $u_1(t\text{-}1)$. These selected variables are exactly the same, as those present in eqn.(2.11).

Table 2-2: List of input candidates with descending order of c_i for Example .1

Variables	*Noise Free Data*		*Noisy Data*	
	C_i	Descending Order of c_i	C_i	Descending Order of c_i
$u_1(t)$	**27.14**	**4**	**26.38**	**4**
$u_1(t\text{-}1)$	**24.64**	**5**	**24.16**	**5**
$u_1(t\text{-}2)$	*14.92*	*10*	*14.66*	*10*
$u_1(t\text{-}3)$	*10.83*	*11*	*9.97*	*11*
$u_1(t\text{-}4)$	*6.17*	*12*	*6.21*	*12*
$u_2(t)$	*2.19*	*16*	*2.29*	*16*
$u_2(t\text{-}1)$	*4.28*	*14*	*4.25*	*14*
$u_2(t\text{-}2)$	*3.70*	*15*	*3.84*	*15*
$u_2(t\text{-}3)$	*5.02*	*13*	*5.66*	*13*
$u_2(t\text{-}4)$	**29.83**	**3**	**29.00**	**3**
$u_2(t\text{-}5)$	*16.38*	*8*	*15.88*	*9*
$u_2(t\text{-}6)$	*18.70*	*7*	*18.22*	*7*
$y(t\text{-}1)$	**43.70**	**1**	**42.60**	**1**
$y(t\text{-}2)$	**33.86**	**2**	**33.32**	**2**
$y(t\text{-}3)$	*22.03*	*6*	*22.30*	*6*
$y(t\text{-}4)$	*16.13*	*9*	*16.87*	*8*

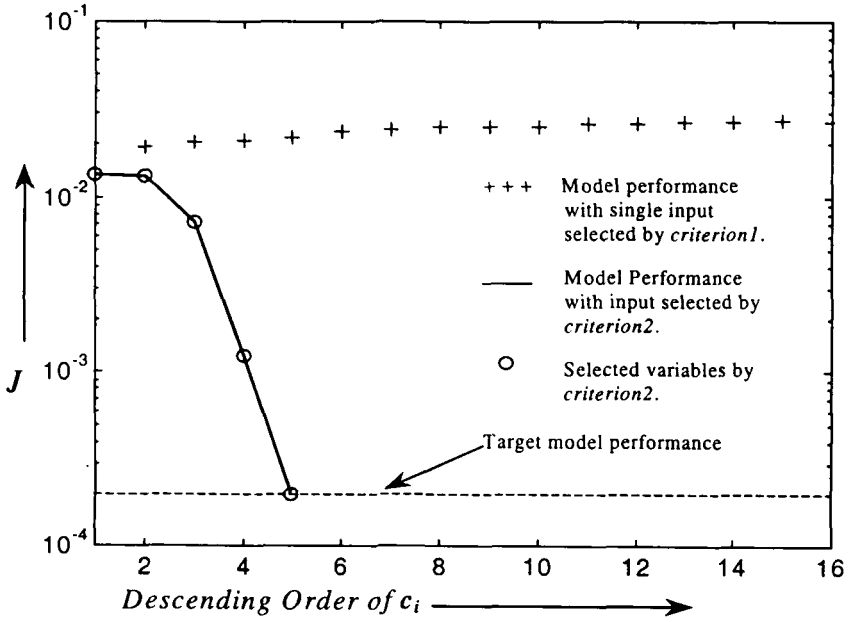


Fig. 2.5: Model Performance vs Descending Order of c_i for Example 2.1 with the noise free data

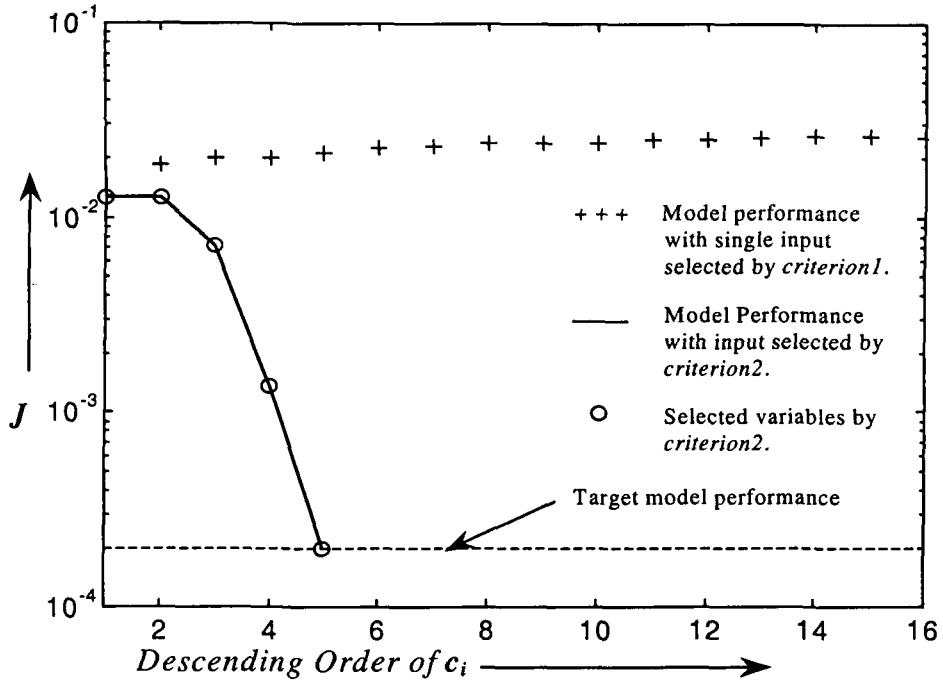


Fig. 2.6: Model Performance vs Descending Order of c_i for Example 2.1 with the noisy data.

Example 2.2

We set the number of fuzzy rules for the model as m=3 from the fuzzy curve approach. In Table 2-3, the values of c_i for each input candidate and the descending order of c_i are listed.

First, 80% of data are used for model learning, and the rest 20% are used for model validation. The model performance with the each individual input candidate is plotted in Fig.2.7. The solid line in Fig 2.7 shows the performance of models with the application of *criterion2*. A circle on the solid line shows the acceptance of that input candidate as input variable and × shows the rejection of that input candidate. The selected variables are $u(t\text{-}5)$, $u(t\text{-}4)$, $y(t\text{-}1)$, $u(t\text{-}3)$ and $y(t\text{-}2)$and the rejected one is $u(t\text{-}6)$. The model performance reaches its target value after adding the input candidate $y(t\text{-}2)$.

Table 2-3: List of input candidates with the descending order of c_i for Example 2.2

Variables	c_i	Descending order of c_i
$u(t)$	*32.91*	*11*
$u(t\text{-}1)$	*42.32*	*10*
$u(t\text{-}2)$	*52.45*	*8*
$u(t\text{-}3)$	**61.23**	**5**
$u(t\text{-}4)$	**66.52**	**2**
$u(t\text{-}5)$	**67.39**	**1**
$u(t\text{-}6)$	*63.95*	*4*
$y(t\text{-}1)$	**65.65**	**3**
$y(t\text{-}2)$	**60.86**	**6**
$y(t\text{-}3)$	*54.28*	*7*
$y(t\text{-}4)$	*47.24*	*9*

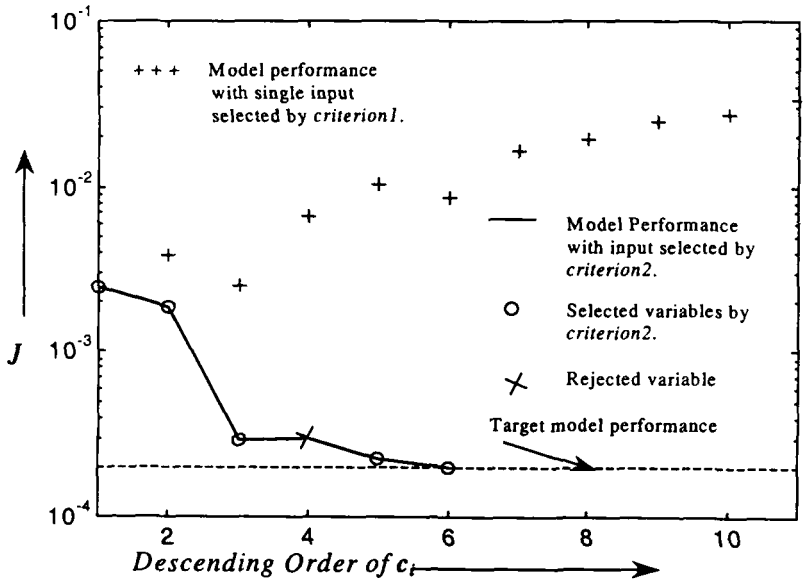


Fig. 2.7: Model Performance vs Descending Order of c_i for Gas furnace data (Example 2.2).

Example 2.3

Here, we set the number of fuzzy rules for the model as m=5 from the fuzzy curve. In Table, 2-4 the values of c_i for each input candidate and the descending order of c_i are listed.

All the data are used for model learning in this example. The model performance with the each individual input candidate is plotted in Fig.2.8. The selected variables are u_3 and u_1. The model performance reaches its target value after adding the input candidate u_1 to u_3

Table 2-4: List of input candidates with the descending order of c_i for Example 2.3

Variables	c_i	Descending order of c_i
u_1	**88.40**	**2**
u_2	*42.13*	*3*
u_3	**89.58**	**1**
u_4	*24.51*	*4*
u_5	*21.78*	*5*

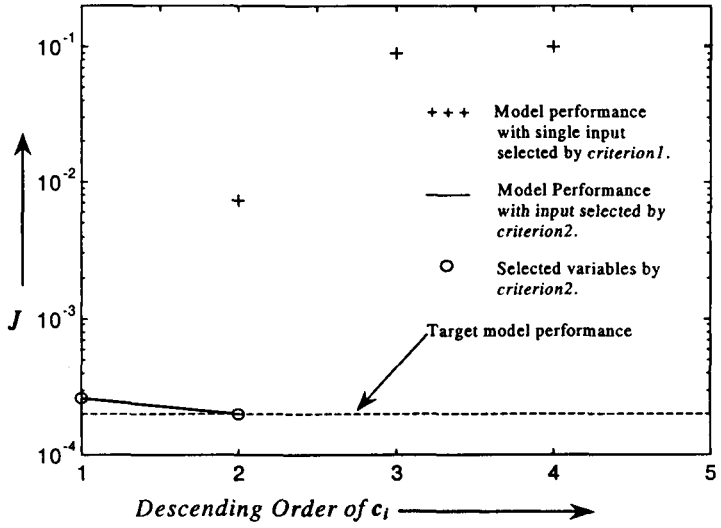


Fig. 2.8: Model Performance vs Descending Order of c_i for Human operation at a chemical plant (Example 2.3).

Example 2.4

We set here the number of fuzzy rules for the model as m=4. Table 2-5 lists the values of c_i for each input candidate and the descending order of c_i.

First, 80% of data are used for model learning, the rest, 20% are used for model validation. The model performance with each individual input candidate is plotted in

Fig.2.9. The solid line in Fig 2.9 shows the performance of models with the application of *criterion2*. The selected variables are x_4, x_2, x_5, x_8, x_{10} and x_1, and the rejected variables are x_3, x_7, x_9 and x_6. Since the model performance never reaches its target, we build the test models with x_9 and x_6, though the minimum performance (greater than the target value) is achieved after adding the variable x_1.

Table 2-5: List of input candidates with the Descending Order of c_i for Example 2.4

Variables	C_i	Descending order of c_i
x_1	**16.98**	**8**
x_2	**40.56**	**2**
x_3	*19.48*	*6*
x_4	**44.23**	**1**
x_5	**28.73**	**3**
x_6	*9.33*	*10*
x_7	*17.19*	*7*
x_8	**25.25**	**4**
x_9	*15.40*	*9*
x_{10}	**23.84**	**5**

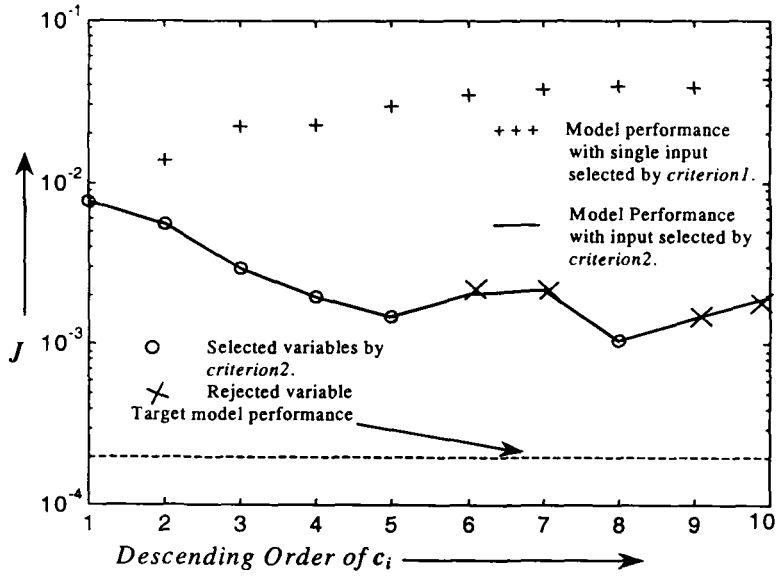


Fig. 2.9: Model Performance vs Descending Order of c_i for Daily price of a Stock market (Example 2.4).

Discussion

It can be observed from Figs.2.5 to.2.9 and shown by + sign, that the input candidate with highest value of c_i gives the minimum model performance when only one candidate is considered as input. It is inferred that the input candidate with the highest value of c_i is a good choice to start building the test models.

The first five candidates are selected without any rejection in Example 2.1 for both noise free data and noisy data as shown in Fig.2.5 and Fig.2.6 respectively. After selecting the first five candidates the input selection procedure is terminated, because the model performance has reached the target value. Similarly, in Fig.2.8 of *Example 2.3* the first two candidates are selected without any rejection and the selection procedure is terminated. In *Example 2.2* five candidates among six are selected with one rejection. Since the model performance has reached the target value, the selection procedure is terminated for the sixth variable. The input selection procedure continues to the last candidate in *Example 2.4* because the model performance does not reach the target value. Six candidates are selected as inputs among ten candidates in this example.

It may be noted here that the performance of the test models of Takagi-Sugeno **[Takagi'85]** and Sugeno-Yasukawa **[Sugeno'93]** has not been compared as it is beyond the scope of thesis.

2-6 Conclusions

In this chapter, fuzzy curve of Lin and CunninghamIII **[Lin'95]** is introduced by proposing AFDM and is interpreted as an unconditional expected value of the output. It is shown that the number of test models is of the order $\mathcal{O}(n^2 p)$ for Takagi and Sugeno

method **[Takagi'85]** and $\mathcal{O}(n^2)$ for Sugeno and Yasukawa method **[Sugeno'93]**. New criteria are proposed for input selection, which yields the number of test models in the order $\mathcal{O}(n)$ and validity of these criteria is demonstrated on the fuzzy models of few dynamic systems.

The performance of Takagi-Sugeno method **[Takagi'85]**, Sugeno-Yasukawa method **[Sugeno'93]** has not been compared with that of proposed approach. Although, this would have given some additional information about how effective they are. It is a matter for further study.

Chapter 3

GENERALIZED ADAPTIVE NEURO-FUZZY INFERENCE SYSTEMS

3-1 Introduction

Fuzzy modeling based on the fuzzy set theory proposed by Zadeh **[Zadeh'73]** has been widely investigated. The aim of the fuzzy modeling is to build fuzzy relations, which are expressed by a set of linguistic propositions derived, either from the experience of a skilled operator or a set of observed input-output data. In the early stages of knowledge based fuzzy logic applications to real life systems, Mamdani has **[Mamdani'74; 77]** used Compositional Rule of Inference (CRI) form of fuzzy model to interpret the operator experience in handling simple operations. However, for some large complex systems, it is almost impossible to establish such knowledge based fuzzy models due to a large number of fuzzy propositions and the highly complicated multidimensional fuzzy relationships. Later, the pioneering work of Takagi and Sugeno **[Takagi'85]** on fuzzy modeling and control has led to several works in the literature **[Xu'87; Fukuda'93; Sugeno'93; Wang'95]** which are termed as multi-model based approaches **[Pedrycz'96].** The basic idea in these approaches is to decompose the complicated input space into subspaces and then approximate the subsystem represented in each subspace by a simple linear regression model. Thus, the overall fuzzy model is considered as a combination of interconnected subsystems

with simpler models and we will refer to this model as TS-model of fuzzy Inference. Using the similar decomposition of the input space, CRI-model interpolates among parallel hyper surfaces perpendicular to the output co-ordinate resulting in a family of hyper surfaces depending upon the fuzziness around the parallel hyper surfaces. On the other hand, TS-model interpolates among the inclined hyper surfaces resulting in a single hyper surface.

The locally tuned and overlap receptive fields were first studied in **[Broomhead'88; Moody'88]**. Local model networks were introduced in **[Moody'89; Jones'89]** and further developed in **[Jones'92]**. The work of Jang-Sun **[Jang'93a]** has proved that under certain conditions, the functional behavior of fuzzy logic and Radial Basis Function (RBF) networks is actually equivalent. This functional equivalence of two models is of great importance for providing learning ability to Fuzzy Inference Systems (FIS) **[Jang'93b; Nauck'97; 98]**. The results of functional equivalence between RBF network and FIS **[Jang'93a]** are restricted to a certain class of TS-model. Later on Hunt et.al., **[Hunt'96]** have generalized this functional equivalence by removing some of the restrictions on the class of RBF networks and fuzzy systems. This generalization is still restricted to Gaussian form of input membership function. Hence the need arises for a generalized membership function catering to a wide class of functions.

Since the Adaptive Network based FIS (ANFIS) is restricted only to TS Model, this chapter attempts to devise a Generalized ANFIS which caters to adaptive learning of both TS-Model and CRI-Model and hence to the proposed Generalized Fuzzy Model (GFM).

This chapter is organized into seven Sections. A brief review of CRI-model and TS-model is given in Section 3-2. This Section includes conditions for

equivalence of CRI-model and TS-model. In Section 3-3, the GFM is proposed using the interpolation property of CRI-model and TS-model among hyper surfaces. Conditions under which GFM can be reduced to CRI-model or TS-model are stated and a Generalized membership function for premise variable is defined in this Section. In Section 3-4 a Generalized Radial Basis Function (GRBF) network is devised. The issue of normalized vs non-normalized network is also tackled in this Section. Section 3-5 deals with the functional equivalence of GFM and GRBF networks along with a brief discussion on the initialization and learning of GRBF network. Simulation results on the examples of chapter 2 are presented in Section 3-6. Finally conclusions are relegated to Section 3-7.

3.2 Fuzzy Models

We now briefly discuss CRI-model and TS-model so as to evolve the conditions under which these two models are equivalent. Subsequently, a Generalized Fuzzy Model (GFM) that exhibits the distinguishing features of both these known models is derived using the interpolation property. Thus, the need for the GFM arises in order to tap the useful features of the CRI and the TS models in one model. A generalized membership function in closed form is proposed for the premise variables.

3-2.1 CRI-model

Each rule of a fuzzy model based on Compositional Rule of Inference (CRI) maps fuzzy subsets in the input space $\boldsymbol{A}^k \subset R^{n_k}$ to a fuzzy subset in the output space $\boldsymbol{B}^k \subset R$, and has the form:

$$R^k : \text{if } x_1 \text{ is } A_1^k \wedge x_2 \text{ is } A_2^k \wedge \ldots \wedge x_{n_k} \text{ is } A_{n_k}^k \text{ then } y \text{ is } \boldsymbol{B}^k \qquad \ldots(3.1)$$

with $k = 1 \ldots m$, m being the number of rules. Each rule is premised on its own input vector $\boldsymbol{x}^k$, where $\boldsymbol{x}^k \subseteq \boldsymbol{x}$; $\boldsymbol{x}$ being the complete system input vector. $\boldsymbol{A}_i^k$ are linguistic labels of fuzzy sets describing the qualitative nature of the input variable x_i, $\wedge$ is a fuzzy conjunction operator (usually of T-norm). $\boldsymbol{B}^k$ are the linguistic labels of fuzzy sets that describe the qualitative state of the output variable y. The firing strength of k^{th} rule, obtained by taking the T-norm (usually min or product operator) of the membership functions of the premise parts of the rule, is:

$$\mu^k(\boldsymbol{x}^k) = \mu_1^k(x_1) \wedge \mu_2^k(x_2) \wedge \ldots \wedge \mu_{n_k}^k(x_{n_k}) \qquad \ldots(3.2)$$

where $\mu_i^k(x_i)$ is the membership function of fuzzy set $\boldsymbol{A}_i^k$. The firing strength of k^{th} rule is also represented as a fuzzy set $\boldsymbol{A}^k \subset R^{n_k}$ in the input space. Hence eqn.(3.1) can be written as

$$R^k : \text{if } \boldsymbol{x}^k \text{ is } \boldsymbol{A}^k \text{ then } y \text{ is } \boldsymbol{B}^k \qquad \ldots(3.3)$$

Let $\phi^k(y)$ be the membership function of fuzzy set $\boldsymbol{B}^k \subset R$ in the output space. $\phi^k(y)$ can be of any shape of convex function type with area v_k and centroid b_k such that

$$\begin{aligned} \text{Area}\left(\boldsymbol{B}^k\right) &= v_k \\ &= \int_y \phi^k(y)\,dy \end{aligned} \qquad \ldots(3.4)$$

and,

$$\begin{aligned} \text{Centroid}\left(\boldsymbol{B}^k\right) &= b_k \\ &= \frac{\int_y y\phi^k(y)\,dy}{\int_y \phi^k(y)\,dy} \end{aligned} \qquad \ldots(3.5)$$

So B^k can be written in functional form as $B^k(b_k, v_k)$. Using T-norm for mapping fuzzy subsets from the input space $A^k \subset R^{n_k}$ to fuzzy subset in the output space $B^k \subset R$, a mapping fuzzy subset B^{*k} is obtained from:

$$\phi^{*k}(y) = \mu^k(x^k) \wedge \phi^k(y) \qquad ...(3.6)$$

S-norm (usually max or sum operator) is used in the output space to join all the mapped region in the output space. Aggregated fuzzy set B^o in the output region is obtained from:

$$B^o = B^{*1} \vee B^{*2} \vee .... \vee B^{*m} \qquad ...(3.7)$$

where, $\vee$ is a fuzzy disjunction operator (usually of S-norm). We apply the weighted average gravity method for defuzzification. The defuzzified output y^o is given by:

$$y^o = \frac{\int_y y\phi^o(y).dy}{\int_y \phi^o(y)dy} \qquad ...(3.8)$$

where, $\phi^o(y)$ is the resultant membership function of $B^o \subset R$ in the output space. Four classes of CRI-model can be derived on the basis of choices present in T-norm and S-norm operators. T-norm is used for mapping the input space to the output space whereas S-norm is used for the aggregation of all the mapped regions in the output space.

Class I: Multiplicative operator for T-norm and Additive operator for S-norm.

Class II: Min operator for T-norm and Additive operator for S-norm.

Class III: Multiplicative operator for T-norm and Max operator for S-norm.

Class IV: Min operator for T-norm and Max operator for S-norm.

For class I and class II models, the discrete version of eqn.(3.8) is reduced to

$$y^o = \sum_{k=1}^{m} \frac{\text{Area}\left(B^{*k}\right).\text{Centroid}\left(B^{*k}\right)}{\sum_{j=1}^{m}\text{Area}\left(B^{*j}\right)} \qquad \ldots(3.9)$$

For class I CRI-model the mapping fuzzy subset $B^{*k}\left(b_k^*, v_k^*\right)$ in the output space for k[th] rule retains the same centroid as that of $B^k\left(b_k, v_k\right)$ but its area v_k^* is $\mu^k\left(x^k\right)$ times v_k irrespective of the shape and support of $\phi^k(y)$. Therefore, $B^{*k}\left(b_k^*, v_k^*\right)$ can be written as $B^{*k}\left(b_k, \mu^k\left(x^k\right)v_k\right)$. From eqn.(3.9), the defuzzified output y^o for class I CRI-model is:

$$y^o = \sum_{k=1}^{m} \frac{\mu^k\left(x^k\right)v_k}{\sum_{j=1}^{m}\mu^j\left(x^j\right)v_j}.b_k \qquad \ldots(3.10)$$

The centroid b_k^* and the area v_k^* of the mapping fuzzy subset $B^{*k}\left(b_k^*, v_k^*\right)$ for class II CRI-model depend on the shape and type of $B^k\left(b_k, v_k\right)$. The centroid b_k^* and area v_k^* are evaluated from the basic principles of geometry. The centroid b_k^* remains the same as the centroid b_k for symmetric functions. To determine the area v_k^*, we consider $\phi^k(y)$ as a symmetric triangular function for simplicity, (See Fig.3.1), given by:

$$\phi^k(y) = \begin{cases} 0 & \text{, for} \quad y \le (b_k - v_k/2) \\ \dfrac{y-(b_k - v_k/2)}{v_k/2} & \text{, for} \quad (b_k - v_k/2) < y \le b_k \\ \dfrac{-y+(b_k + v_k/2)}{v_k/2} & \text{, for} \quad b_k < y < (b_k + v_k/2) \\ 0 & \text{, for} \quad (b_k + v_k/2) < y \end{cases} \qquad \ldots(3.11)$$

We have considered the triangular function for the consequent part for ease of computation of area. The shaded region shows the mapped fuzzy set B^{*k} in Fig.3.1 and its membership function $\phi^{*k}(y)$ is:

$$\phi^{*k}(y)=\begin{cases} 0 & \text{, for} \quad y\le(b_k-v_k/2) \\ \dfrac{y-(b_k-v_k/2)}{v_k/2} & \text{, for} \quad (b_k-v_k/2)<y\le(b_k-r_k/2) \\ \mu^k(x^k) & \text{, for} \quad (b_k-r_k/2)<y\le(b_k+r_k/2) \\ \dfrac{-y+(b_k+v_k/2)}{v_k/2} & \text{, for} \quad (b_k+v_k/2)<y<(b_k+v_k/2) \\ 0 & \text{, for} \quad (b_k+v_k/2)<y \end{cases} \qquad ...(3.12)$$

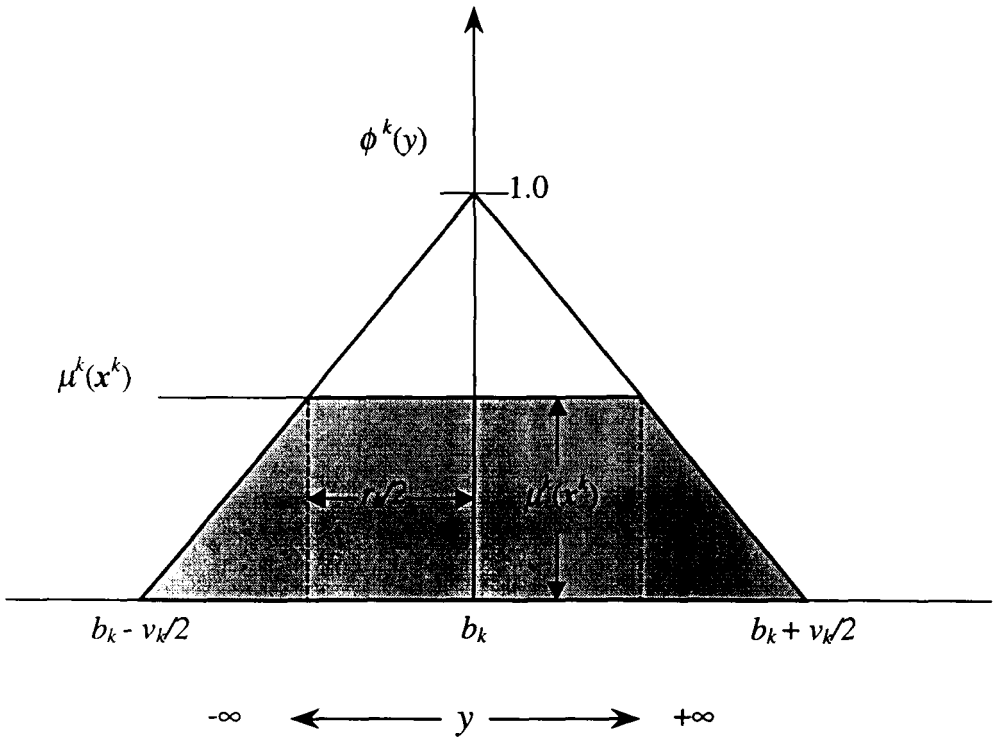


Fig 3.1. Membership function $\phi^k(y)$ (white upper triangle and gray trapezoid) and $\phi^{*k}(y)$ (only gray Trapezoid) corresponding to $\boldsymbol{B}^k$ and $\boldsymbol{B}^{*k}$ for the k^{th} rule of class II CRI-model respectively.

Upper width r_k of shaded trapezoidal portion of Fig.3.1 is found from the relation

$$\mu^k(x^k)=\begin{cases} \phi^k(b_k-r_k/2) & \text{, for} \quad (b_k-v_k/2)<y\le b_k \\ \phi^k(b_k+r_k/2) & \text{, for} \quad b_k<y<(b_k+v_k/2) \end{cases}$$

From the above relation, we obtained

$$\mu^k(x^k) = \frac{(b_k - r_k/2) - (b_k - v_k/2)}{v_k/2}$$

$$\Rightarrow \quad \mu^k(x^k) = \frac{v_k - r_k}{v_k}$$

$$\Rightarrow \quad r_k = \left[1 - \mu^k(x^k)\right].v_k$$

Hence, the area of shaded trapezoidal region, v_k^*, of Fig.3.1 is:

$$\begin{aligned}\text{Area}(B^{*k}) &= v_k^* \\ &= \text{height} \times \frac{1}{2}[\text{Lower base} + \text{Upper base}] \\ &= \frac{1}{2}\mu^k(x^k)[v_k + r_k] \\ &= \frac{1}{2}\mu^k(x^k)\left[2 - \mu^k(x^k)\right].v_k\end{aligned}$$

The defuzzified output y^o for class II CRI-model from eqn.(3.9) is:

$$\begin{aligned}y^o &= \sum_{k=1}^{m} \frac{\mu^k(x^k)\left[2 - \mu^k(x^k)\right].v_k}{\sum_{j=1}^{m} \mu^j(x^j)\left[2 - \mu^j(x^j)\right].v_j}.b_k \\ &= \sum_{k=1}^{m} \frac{\hat{\mu}^k(x^k).v_k}{\sum_{j=1}^{m} \hat{\mu}^j(x^j).v_j}.b_k\end{aligned} \qquad \ldots(3.13)$$

where, $\hat{\mu}^k(x^k) = \mu^k(x^k)\left[2 - \mu^k(x^k)\right]$ is the processed firing strength of k^{th} rule for class II CRI-model, with symmetric triangular membership function of fuzzy set B^k. In the next Section, a choice of the consequent membership function and a justification for the symmetric Gaussian and triangular function from the centroid b_k and the area v_k will be made.

In eqn.(3.10) and eqn.(3.13) we can see that v_k is a weight to the firing strength (for class I) or processed firing strength (for class II) of a rule before its

normalization. Hence we define v_k as the index of fuzziness of the consequent membership function B^k.

3-2.2 Takagi-Sugeno Model

The Takagi-Sugeno (TS) model was introduced in [Takagi'85] as a hybrid model, which integrates the fuzzy conditions in the input space with the functional relationships in the output space. In this model, the premise part of the rule is the same as that of eqn.(3.1) or eqn.(3.3) but only the consequent part is different. Instead of fuzzy sets as in the output space of CRI-model, TS-model has a linear or nonlinear relationship of inputs $f^k(x^k)$ in the output space. Rules of TS model are of the following form:

$$R^k : \text{if } x^k \text{ is } A^k \text{ then } y \text{ is } f^k(x^k) \qquad ...(3.14)$$

A linear form of $f^k(x^k)$ in eqn. (3.14) is as follows:

$$f^k(x^k) = b_{k0} + b_{k1}x_1 + \ldots + b_{kn_k}x_{n_k} \qquad ...(3.15)$$

where, $f^k(x^k)$ defines a locally valid model on the support of the Cartesian product of fuzzy sets constituting the premise parts. The firing strength of each rule is calculated using (3.2). The normalized firing strength for the normalized calculation or non-normalized firing strength for the non-normalized calculation is then multiplied with the output function $f^k(x^k)$. The normalized form of the overall output of the TS-model is defined as:

$$y^o = \sum_{k=1}^{m} \frac{\mu^k(x^k)}{\sum_{j=1}^{m} \mu^j(x^j)} \cdot f^k(x^k) \qquad ...(3.16)$$

and the non-normalized form of the overall output of the TS-model is defined as:

$$y^o = \sum_{k=1}^{m} \mu^k\left(x^k\right) f^k\left(x^k\right) \qquad \text{...(3.17)}$$

We defer the discussion on the relative merits of normalized and non-normalized forms to the subsequent Section.

3-2.3 Conditions for the Equivalence of CRI and TS Models

To establish a relation between CRI-model and TS-model, class I CRI-model is considered for which $\hat{\mu}^k\left(x^k\right) = \mu^k\left(x^k\right)$. Comparison of eqns.(3.10) and (3.16) indicates that these two models are functionally equivalent under the following conditions:

1. Multiplicative T-norm should be used for mapping fuzzy subsets from the input space $\boldsymbol{A}^k \subset R^{n_k}$ to fuzzy subsets in the output space $\boldsymbol{B}^k \subset R$.
2. Additive S-norm should be used for obtaining $\boldsymbol{B}^o$ in the output space.
3. $v_1 = v_2 = \ldots = v_m$
4. $f^k\left(x^k\right) = b_k$.

With the condition (4), $f^k\left(x^k\right)$ turns out to be a singleton in TS-model and with the condition (3), fuzziness of all the consequent variable membership functions becomes equal in CRI-model. We now propose a Generalized form of Fuzzy Model (GFM), encompassing the property of CRI-model, i.e., possessing varying fuzziness of consequent variable membership functions, and that of TS-model, i.e., possessing the consequent part as a local model, which is a function of premise variables.

3-3 Generalized Form of Fuzzy Model

CRI-model inhibits the property of fuzziness around the fixed centroid of consequent while TS-model gives a varying singleton for the consequent part in each fuzzy rule. To combine both these properties we may use a rule of the form:

$$R^k : \text{if } x^k \text{ is } A^k \text{ then } y \text{ is } B^k\left(f^k\left(x^k\right), v_k\right) \qquad ...(3.18)$$

The earlier four classes of CRI-model are also applicable here. Figure 3.1(A) depicts a two-rule two-input FIS to show four classes of GFM. To reduce these four classes of GFM into CRI models the centroids of the consequent membership functions must be constants.

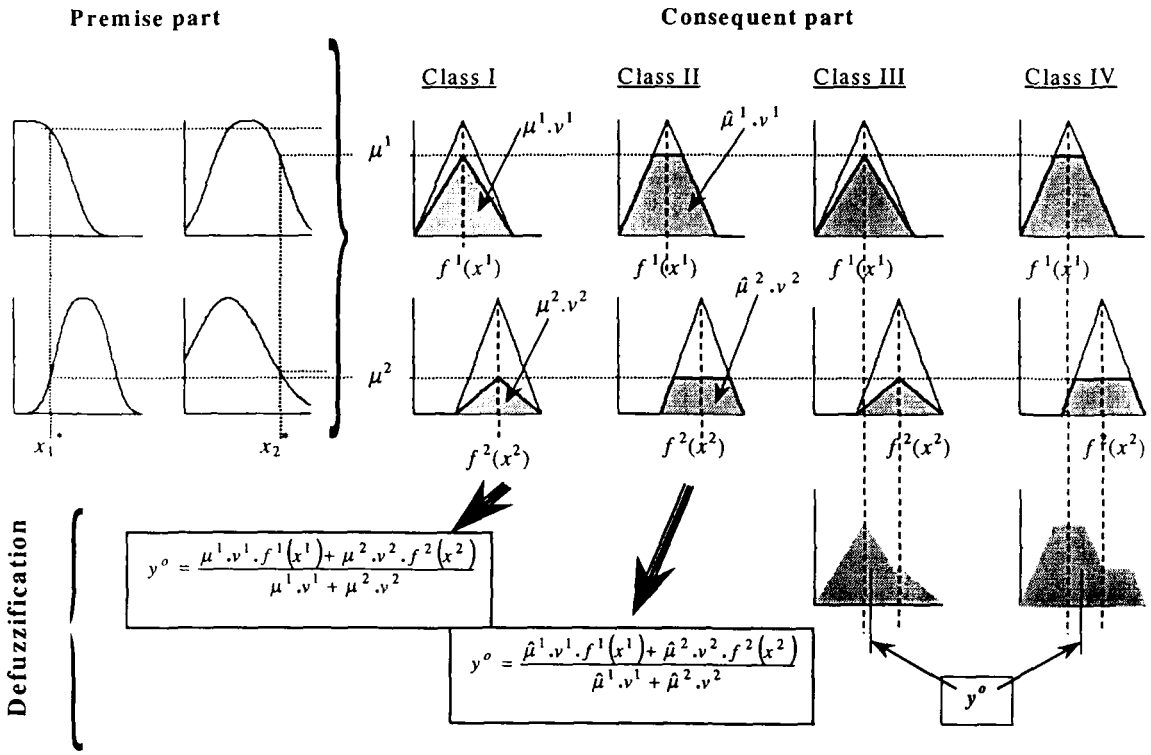


Fig 3.1(A). Four classes of GFM under two types of T-norm (i.e. product and min) and S-norm (i.e., sum and max).

Corresponding to the class I CRI-model, the defuzzified output y^o for the GFM under multiplicative T-norm operator defined by eqn.(3.6), additive S-norm operator defined by eqn.(3.7), is given by:

$$y^o = \sum_{k=1}^{m} \frac{\mu^k\left(x^k\right) v_k}{\sum_{j=1}^{m} \mu^j\left(x^j\right) v_j} . f^k\left(x^k\right) \qquad \ldots(3.19)$$

Corresponding to the class II CRI-model, the defuzzified output y^o for the GFM under min T-norm operator defined by eqn.(3.6), additive S-norm operator defined by eqn.(3.7), is given by:

$$\begin{aligned} y^o &= \sum_{k=1}^{m} \frac{\mu^k\left(x^k\right)\left[2-\mu^k\left(x^k\right)\right] . v_k}{\sum_{j=1}^{m} \mu^j\left(x^j\right)\left[2-\mu^j\left(x^j\right)\right] v_j} . f^k\left(x^k\right) \\ &= \sum_{k=1}^{m} \frac{\hat{\mu}^k\left(x^k\right) v_k}{\sum_{j=1}^{m} \hat{\mu}^j\left(x^j\right) v_j} . f^k\left(x^k\right) \end{aligned} \qquad \ldots(3.20)$$

The class II GFM-model can be reduced to the class I GFM-model, with $\hat{\mu}^k\left(x^k\right) = \mu^k\left(x^k\right)$ (conditions 1) and the class I GFM-model can be reduced to, either CRI-Model by applying the condition 4, or TS-Model by applying the condition 3, as well as enforcing the condition 2. GFM is associated with normalized calculation. For non-normalized calculation, area v_k can be absorbed into the function $f^k\left(x^k\right)$ so that Generalized form will be reduced to TS-model. For non-normalized calculation, CRI-model becomes a special case of TS-model. We now present an example that illustrates the effect of index of fuzziness v_k on interpolation among the local models.

Example 3.1:

Here we consider two GFM rules and show how they can be interpolated. By varying index of fuzziness, fuzzy rules interpolate the overall output between the local models in many ways. This is the advantage of the GFM over TS and CRI model. Suppose we have two rules as under.

$$R^1 : \text{if } x \text{ is } \mu^1(x) \text{ then } y \text{ is } B^1\left(f^1(x) = 0.2x + 9, v_1\right)$$

$$R^2 : \text{if } x \text{ is } \mu^2(x) \text{ then } y \text{ is } B^2\left(f^2(x) = 0.6x + 2, v_2\right)$$

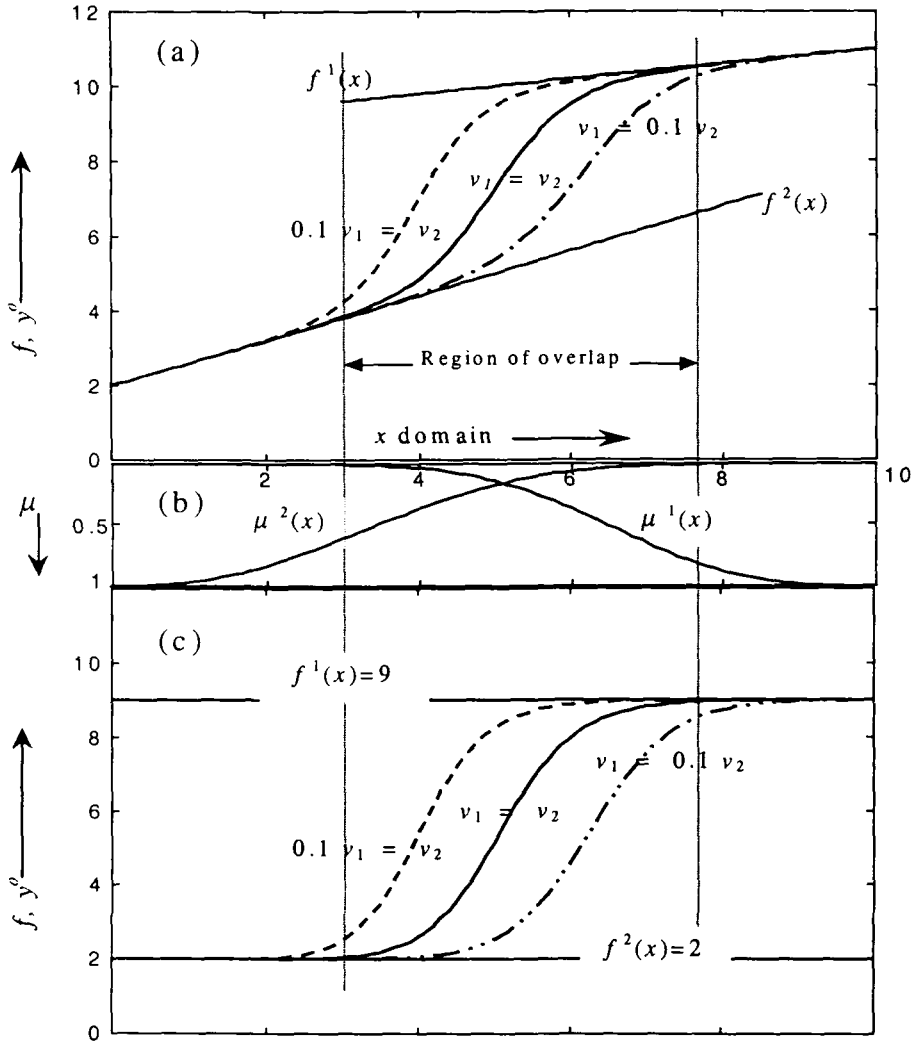


Fig 3.2. Effect of v_k on overall defuzzified output

Figure 3.2 shows the graphs of $f^1(x)$, $f^2(x)$, $\mu^1(x)$, and $\mu^2(x)$, as a function of x, along with the relationships between x and y derived from the rules R^1 and R^2 for three cases (i) $v_1 = v_2$ shown by solid line, (ii) $v_1 > v_2$ shown by dashed line and (iii)

$v_1 < v_2$ shown by dashed-dot line. In this example, the premise part constitutes a single input, but the results can be easily extended to the multi-input case. The effect of index of fuzziness v_k is prominent only over the region of overlap of premise fuzzy sets. It can be seen that interpolation takes place between the consequent (local model) parts of rules, i.e., $f^1(x)$ and $f^2(x)$, according to their index of fuzziness over the region of overlap of the premise fuzzy sets $\mu^1(x)$, and $\mu^2(x)$ as shown in Fig. 3.2(b). Interpolation moves towards that consequent which has a higher value of index of fuzziness v_k. By associating the index of fuzziness to the consequent parts for a fixed region of overlap of the premise membership functions, the fuzzy rules interpolate the overall output between the local models in many ways instead of unique way as shown by the solid line for TS-model depicted in Fig. 3.2(a).

If $f^1(x) = b_1 = 9$ in R^1 and $f^2(x) = b_2 = 2$ in R^2 then we obtain CRI model in which interpolation results in a family of curves between two parallel lines with $f^1(x) = 9$ and $f^2(x) = 2$ due to different v_1 and v_2 as shown in Fig 3.2(c)

3-3.1 Choice of premise parameters

With multiplicative T-norm for the premise part of the rule, Gaussian function is the right choice for A_i^k. Additionally we need a varying shape from triangular to trapezoidal, so we choose generalized Gaussian type functions for A_i^k defined as:

$$\mu_i^k(x_i) = \exp\left(-\left|\frac{x_i - c_{ki}}{\sigma_{ki}}\right|^{l_{ki}}\right) = \exp\left(-\left|a_{ki}(x_i - c_{ki})\right|^{l_{ki}}\right) \qquad ...(3.21)$$

The indices i, k indicate i^{th} input x_i, and k^{th} rule respectively. c_{ki} is the central value of fuzzy set for i^{th} premise variable x_i corresponding to k^{th} rule, σ_{ki} or $(1/a_{ki})$ represents the width of fuzzy set, l_{ki} controls the shape of fuzzy set. This function

can approximate the symmetric triangular to trapezoidal membership functions depending on the value of l_{ki}. From the definition of A_i^k in eqn.(3.21) it can be written in the functional form as $A_i^k(c_{ki}, a_{ki}, l_{ki})$. Figure 3.3 shows the shapes of premise fuzzy membership functions for $l = 0.5, \ldots, 5.0$.

For the Gaussian type membership function under multiplicative T-norm, the premise truth value or firing strength or the membership function of the premise fuzzy set A^k of the k^{th} rule is :

$$\mu^k(x^k) = \prod_{i=1}^{n_k} \exp\left(-\left| \frac{x_i - c_{ki}}{\sigma_{ki}} \right|^{l_{ki}} \right) = \exp\left(-\sum_{i=1}^{n_k} \left| \frac{x_i - c_{ki}}{\sigma_{ki}} \right|^{l_{ki}} \right) \quad \ldots(3.22)$$

$$\text{or} \quad = \prod_{i=1}^{n_k} \exp\left(-|a_{ki}(x_i - c_{ki})|^{l_{ik}} \right) = \exp\left(-\sum_{i=1}^{n_k} |a_{ki}(x_i - c_{ki})|^{l_{ki}} \right)$$

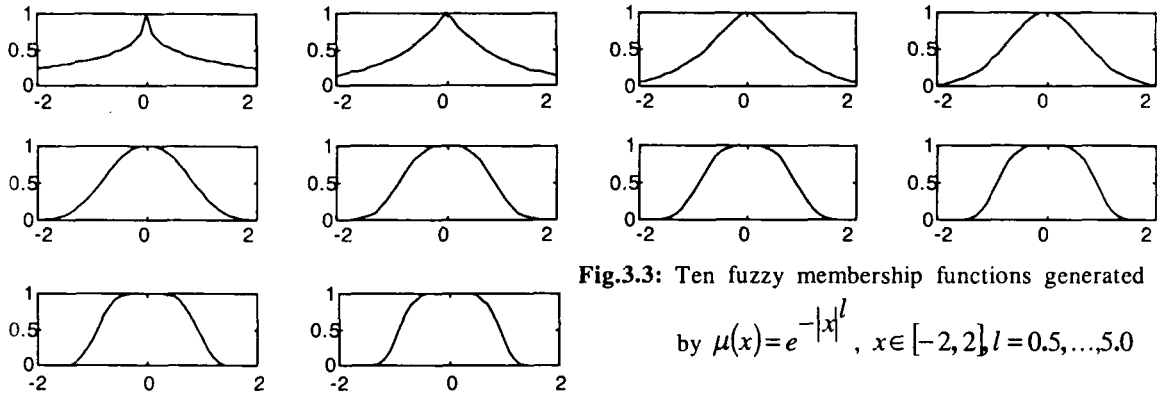


Fig.3.3: Ten fuzzy membership functions generated by $\mu(x) = e^{-|x|^l}$, $x \in [-2, 2], l = 0.5, \ldots, 5.0$

3-2.2 Choice of consequent parameters

From the model (defuzzified) output, it can be inferred that only two properties, viz., centroid b_k and index of fuzziness v_k of consequent membership function $B^k(b_k, v_k)$ are of great importance. As such B^k can also be expressed in terms of two to five parameters as listed in Table 3.1 depending upon the choice of its membership function.

Table 3.1: Different functions and their parameters

Types of Mem. fun.		Symmetric		Unsymmetric	
		No.	Function	No.	Function
(a)	**Triangular**	2	$B^k(y: b_k, v_k)$	3	$B^k(y; a_k, b_k, c_k)$
(b)	**Gaussian**	2	$B^k(y: c_k, \sigma_k)$	3	$B^k(y: c_k, \sigma_{k1}$ *for* $y \leq c_k$ σ_{k2} *for* $y > c_k)$
(c)	**Trapezoidal**	3	$B^k(y: b_k, d_{k1}, d_{k2})$	4	$B^k(y; a_k, b_k, c_k, d_k)$
(d)	**Gen. Gaussian**	3	$B^k(y: c_k, \sigma_k, l_k)$	5	$B^k(y: c_k, \sigma_{k1}, l_k$ *for* $y \leq c_k$ σ_{k2}, l_{k2} *for* $y > c_k)$

The parameters of symmetric triangular membership function are the center b_k and width $v_k/2$ since the height is unity. The parameters of unsymmetric triangular membership function are the unsymmetric point b_k, the left hand side zero a_k, and the right hand side zero c_k.

The parameters of symmetric trapezoidal membership function are the center b_k, the distance from the center corresponding to the constant value of membership, (i.e., '1'), d_{k1}, and the distance from the center corresponding to zero membership, d_{k2}. The parameters of unsymmetrical trapezoidal membership function are the left hand side zero a_k, the right hand side zero d_k, the right side high value point b_k, and the left side high value point c_k.

The parameters of symmetric Gaussian membership function are the center c_k, and σ_k , which is the representative of Width of Gaussian type Funtion(WGTF) of Fuzzy Set. Here onwards, σ_k will be referred to as WGTF. The parameters of unsymmetric Gaussian membership function are the unsymmetric point c_k, the left-hand side WGTF σ_{k1}, and the right hand side WGTF σ_{k2}.

The parameters of symmetric generalized Gaussian membership function are the center c_k, the WGTF σ_k and power l_k. The parameters of unsymmetric generalized

Gaussian membership function are the unsymmetric point c_k , the left hand side WGTF σ_{k1} with power l_{k1}, and the right hand side WGTF σ_{k2} with power l_{k2}.

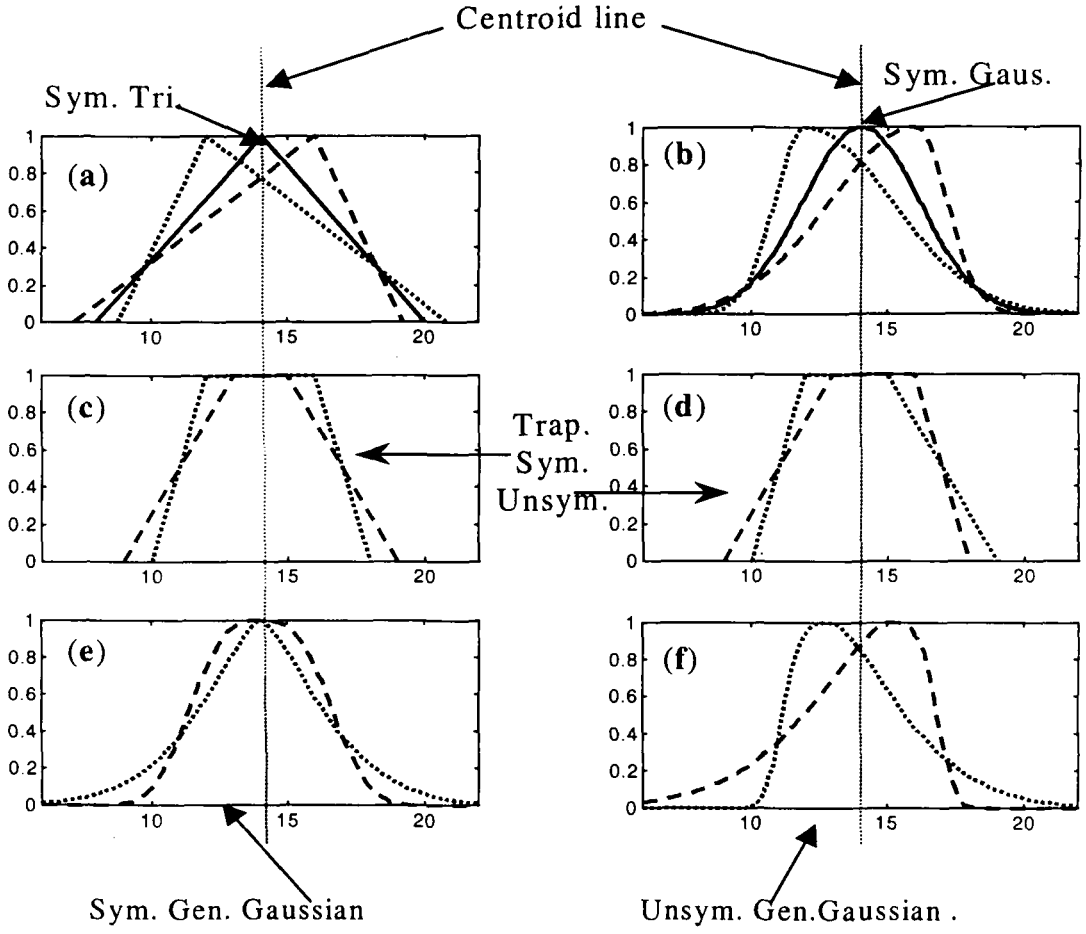


Fig. 3.4: Different choices of consequent membership functions for the same centroid and area.

For a given type of $\boldsymbol{B}^k$ and its parameters, a unique centroid b_k and index of fuzziness v_k can be evaluated. Alternatively, given b_k and v_k, $\boldsymbol{B}^k$ can only be uniquely evaluated for symmetric triangular and symmetric Gaussian functions. For other types listed in Table 3.1, the parameters are not unique. In general, from the given values of b_k and v_k unique $\boldsymbol{B}^k$ can only be determined for a specific type represented by two parameters. On the other hand, with three or more parameters, $\boldsymbol{B}^k$ can not be uniquely determined for given value of b_k, and v_k. Figure 3.4 shows

shapes of membership functions listed in Table 3.1 for fixed values of the centroid and the area. It may be seen that finding parameters of unsymmetric function from the area is more difficult. It involves a lot of computations.

3-4 RBF Networks

As a means to learn the GFM, we describe the Generalized Radial Basis Function (GRBF) network and discuss the issues related to the normalized and the non-normalized networks. The conditions under which it reduces to the Hunt's RBF and Standard RBF networks are then given. The main features of standard networks need not be discussed. Instead, we present the main features of GRBF networks, which are more general. The main features of the GRBF network are:

1. Each processing unit in the network receives all the inputs during the learning phase of the network to keep the number of parameters the same at all input nodes. After learning, some of the inputs may be removed from the premise parts of fuzzy if-then rules if the learned membership function value of the corresponding variable remains almost constant, i.e., nearly "1". Some of the fuzzy if-then rules may be removed if the learned membership function value of any of the variable remains nearly "0"over its domain (See *Examples 3.3* and *3.4*). For this purpose we use a generalized membership function for the input variable. This is important as not all elements of the input vector need necessarily appear in the premise part of the corresponding fuzzy if-then rules. However, all system inputs are considered in vector $\boldsymbol{x}$ while feeding to the network.
2. The GRBF processing units in the network have differing widths and powers (instead of "2" as in Gaussian) in each dimension of the input vector. This results

in differing shapes and sizes of the corresponding membership functions in different dimensions.

3. Normalizing units in the network normalize the weighted outputs from the corresponding processing units by dividing them by the sum of all weighted outputs from all the processing units. This is necessary for the network using normalized computation as explained later.
4. The output links of the network consist of local models, possibly nonlinear, which process the network inputs. The standard RBF network has constant weights on the links connected to the output unit.

The above features of GRBF network aim at the learning of GFM. We have discussed above how the GRBF network is functionally similar to GFM. Whatever the operations required to determine GFM are all incorporated in the GRBF network. Next, we see the architecture of GRBFN.

3-4.1 Architecture of Generalized Radial Basis Function (GRBF) Network

Figure 3.5 shows the four layered architecture of GRBF network. This four layered connectionist structure effectively performs the fuzzy inferencing with some minor restrictions. Layer 1, layer 2 and layer 3 have m nodes each corresponding to each rule. There is only one node in layer 4.

Layer 1: The nodes in this layer represent generalized radial basis functions. Each node in this layer has exactly n inputs and is a special type of radial basis function. The output of each basis function node, $\varphi^k(x)$ will be defined later in this Section.

Layer 2: Each node in layer 2 operates on its input by a function **P**, yielding the output given by:

$$\hat{\varphi}^k(x) = \mathbf{P}\{\varphi^k(x)\}. \qquad \ldots(3.23)$$

This function **P** is unary for class I GFM, and it has the following form for class II GFM:

$$\mathbf{P}\{\varphi^k(x)\} = \varphi^k(x)(2 - \varphi^k(x))$$

Each node of layer 2 provides a transformation function which may be a unary function or function defined as $p(u) = u.(2\text{-}u)$.

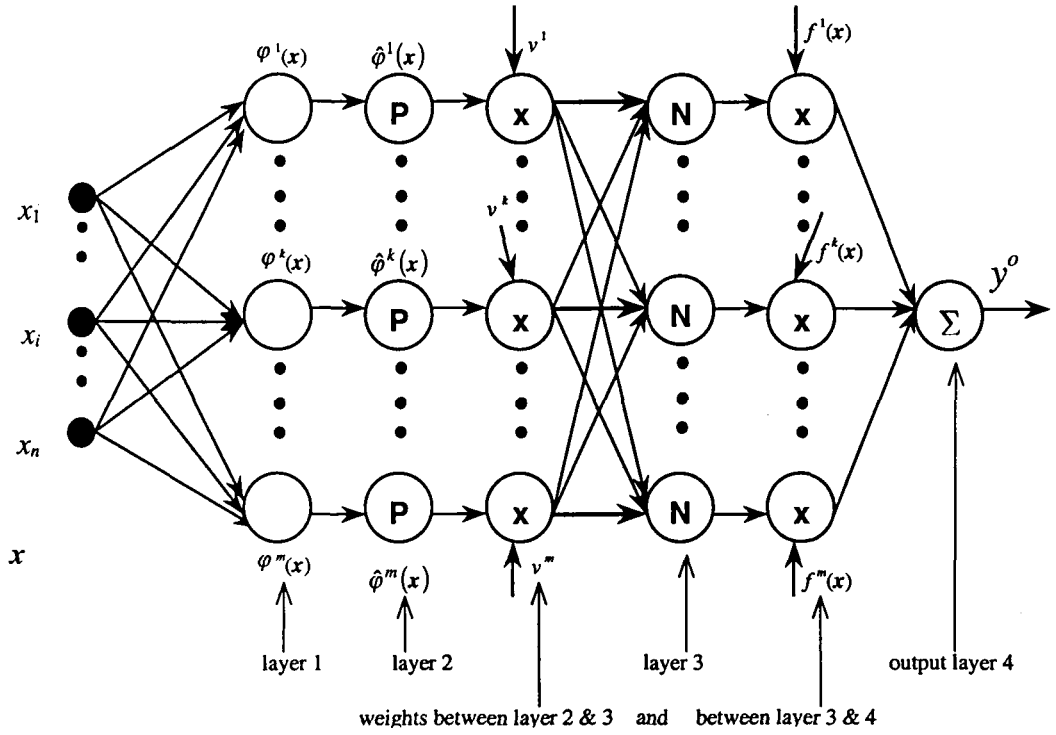


Fig. 3.5: The Architecture of GRBF network for normalized calculation.

Layer 3: The nodes in layer 3 normalized its input coming from layer 2. All nodes in this layer are shown by **N**. Each node has m inputs, i.e. $\hat{\varphi}^k(x)$, $(k = 1, \ldots, m)$, which are the weighted outputs of nodes in layer 2 with weights v_k. The k^{th} node of layer 3 calculates its output as the ratio between the weighted output of k^{th} node in layer 2 and the sum of the

weighted outputs of all nodes in layer 2. The output of each node in layer 3 $\overline{\hat{\varphi}^k(x).v_k}$ is therefore :

$$\overline{\hat{\varphi}^k(x).v_k} = \frac{\hat{\varphi}^k(x).v_k}{\sum_{k=1}^{m} \hat{\varphi}^k(x).v_k} \qquad ...(3.24)$$

There is only one parameter v_k to be adjusted in this layer.

Layer 4: There is only one node in the output layer. Input to this node comes from the outputs of all nodes of layer 3. It performs the summation of all inputs weighted with local model $f^k(x)$ giving rise to the output as:

$$y^o = \sum_{k=1}^{m} \overline{\hat{\varphi}^k(x).v_k} . f^k(x)$$

$$= \sum_{k=1}^{m} \frac{\hat{\varphi}^k(x).v_k}{\sum_{k=1}^{m} \hat{\varphi}^k(x).v_k} . f^k(x) \qquad ...(3.25)$$

This basis function network represents a generalized form of network. Since the weights $f^k(x)$ are functions of the input vector x, a general form of basis function $\varphi^k(x)$ is used as an activation function whose processed-weighted-normalized value $\overline{\hat{\varphi}^k(x).v_k}$ defines the validity of $f^k(x)$. Thus the network, through its local activation function $\varphi^k(x)$, partitions the input space into *m* operating regions on each of which a local model $f^k(x)$ acts with weight v_k, which represents fuzziness around $f^k(x)$. The network smoothly joins these local models together through interpolation to form the overall global model $f(x)$ as discussed in Section 3-3.

Recall that the generalized membership function defined earlier has three parameters for each premise variable. In each basis function there are three sets of parameters; the center vector $\boldsymbol{c}_k \in R^n$, width vector $\boldsymbol{a}_k \in R^n$ and power vector $\boldsymbol{l}_k \in R^n$. We may define these vectors as:

$$\boldsymbol{x} = \{x_1, x_2, \ldots, x_n\}' \qquad \ldots(3.26)$$

$$\boldsymbol{c}_k = \left\langle \{c_{k1}, c_{k2}, \ldots, c_{kn}\}';\ \forall c_{ki} \in R \right\rangle \qquad \ldots(3.27)$$

$$\boldsymbol{a}_k = \left\langle \{a_{k1}, a_{k2}, \ldots, a_{kn}\}';\ \forall a_{ki} \in R^+ \right\rangle$$

$$\text{or} \quad = \left\langle \{1/\sigma_{k1},\ 1/\sigma_{k2}, \ldots, 1/\sigma_{kn}\}';\ \forall \sigma_{ki} \in R^+ \ \text{ and } \ > 0 \right\rangle \qquad \ldots(3.28)$$

$$\boldsymbol{l}_k = \left\langle \{l_{k1}, l_{k2}, \ldots, l_{kn}\}';\ \forall l_{ki} \in R^+ \right\rangle \qquad \ldots(3.29)$$

In view of definitions 1 to 3 of Appendix A and the vectors in eqn.(3.26) to (3.29), we may define a generalized distance vector from the center $\boldsymbol{c}_k$ to a point $\boldsymbol{x}$ as

$$\boldsymbol{d}_k = \left\| \boldsymbol{x} - \boldsymbol{c}_k \right\| .\wedge \left(0.5 * \boldsymbol{l}_k \right) \qquad \ldots(3.30)$$

The width matrix is

$$\Delta_k \quad = \mathbf{DM}\left(\boldsymbol{a}_k .\wedge \boldsymbol{l}_k \right)$$

$$\text{or} \quad = \begin{bmatrix} \left(\frac{1}{\sigma_{k1}}\right)^{l_{k1}} & 0 & \ldots & 0 \\ 0 & \left(\frac{1}{\sigma_{k2}}\right)^{l_{k2}} & \ldots & 0 \\ \cdot & \cdot & \ldots & \cdot \\ \cdot & \cdot & \ldots & \cdot \\ \cdot & \cdot & \ldots & \cdot \\ 0 & 0 & \ldots & \left(\frac{1}{\sigma_{kn}}\right)^{l_{kn}} \end{bmatrix} \qquad \ldots(3.31)$$

Now using eqn.(3.30) and eqn.(3.31), a GRBF is defined as follows:

$$\varphi^k(\boldsymbol{x}) = \exp(-d_k' \, \Delta_k \, d_k)$$

or $$= \exp\left[-\left(\left\{\|\boldsymbol{x}-\boldsymbol{c}_k\|.\wedge(0.5*\boldsymbol{l}_k)\right\}'.[\mathbf{DM}(\boldsymbol{a}_k .\wedge \boldsymbol{l}_k)]\left\{\|\boldsymbol{x}-\boldsymbol{c}_k\|.\wedge(0.5*\boldsymbol{l}_k)\right\}\right)\right]$$

or $$= \exp\left[-\left\{\left|\frac{x_1-c_{k1}}{\sigma_{k1}}\right|^{l_{k1}} + \left|\frac{x_2-c_{k2}}{\sigma_{k2}}\right|^{l_{k2}} + \ldots + \left|\frac{x_n-c_{kn}}{\sigma_{kn}}\right|^{l_{kn}}\right\}\right]$$

...(3.32)

This basis function is generalized in the sense that the basis functions of Hunt have the same power, i.e., "2", and the standard radial basis functions have the same width besides the same power in each dimension. The GRBF's are defined on hyper patches in the input space (because of different widths and powers in each dimension) as opposed to hyper ellipsoids in Hunt's basis function and hyper spheres in Standard Gaussian RBF's.

To clarify further we may note that the standard radial basis functions have the same width and the same power, the basis function of Hunt have the same power but different width, and the generalized basis function have different power and different widths in each dimension.

In a similar way we can define the parameters of local models as

$$f^k(\boldsymbol{x}^k) = \boldsymbol{b}_k'.[1 \quad \boldsymbol{x}'] \qquad \text{...(3.33)}$$

where $\boldsymbol{b}_k$ is given by

$$\boldsymbol{b}_k = \{b_{k0}, b_{k1}, \ldots, b_{kn}\}' \qquad \text{...(3.34)}$$

k^{th} GRBF link parameter vector $\boldsymbol{s}_k$ is defined as:

$$\boldsymbol{s}_k = \{\boldsymbol{c}_k', \boldsymbol{a}_k', \boldsymbol{l}_k', \boldsymbol{b}_k', v_k\}' \qquad \text{...(3.35)}$$

Finally the GRBF network is represented by the parameter matrix $\boldsymbol{S}$, which is defined as:

$$\boldsymbol{S} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_m\} \qquad \text{...(3.36)}$$

3-4.2 Normalized network Vs non-normalized network

There are two forms of GRBFN employing either the normalized computation or non-normalized computation. We will now discuss their relative merits. Before going to state the conditions under which GRBF network can be reduced to Hunt's network and standard RBF network, need arises to stress upon the distinguishing features of Normalized network over the non-normalized network. A normalized form of network is similar to that of Hunt's RBF network as shown in Fig.3.6. Shorten and Murray-Smith **[Shorten'94]** have reported some of the side effects on the use of normalized basis functions. These side effects in fact go in favor of normalized network. Approximation capabilities of normalized network have been demonstrated in **[Beniam'94]**, and it has an edge over the non-normalized network as shown subsequently in *Example 3.2.* In the normalized network $\varphi^k(\boldsymbol{x})$ are the non-normalized homogeneous functions. For achieving the same model performance with the non-normalized network, the basis function $\varphi^k(\boldsymbol{x})$ should be in the normalized form. It is difficult to represent the normalized basis functions in both continuos and homogeneous forms (only possible by spline function). In the proposed normalized network normalization is done after weighting $\varphi^k(\boldsymbol{x})$ by v_k. Some of the new results excluding those in **[Shorten'94]** regarding the use of the normalized network are summarized below:

i) *Loss of independence*: A change in v_k or any of the parameters of $\varphi^k(\boldsymbol{x})$ not only affects its own weighted normalized basis function $\overline{\varphi^k(\boldsymbol{x})v_k}$, but also the rest of the weighted normalized basis functions.

ii) *Change of shape*: A change in v_k or any of the parameters of $\varphi^k(x)$ in the proximity of any basis function leads to a corresponding change in the shape of its weighted normalized basis function.

iii) *Covering of input space*: Weighted normalized basis function covers the whole input space instead of a part of it.

iv) *Shift in maxima*: Shift in maxima occurs because of difference in parameters σ (WGTF) and /or l (power) of neighboring functions (See Fig. 3.7a). The values of maxima depend upon the overlap among the neighboring basis functions and the corresponding v_k's.

v) *Reactivation*: This occurs in the hollow region of hyperspace, where the basis functions are not defined, due to the difference in parameters σ and/or l among the neighboring basis functions in the proximity of hollow regions. In the absence of reactivation, hollow regions may be covered by the normalized basis functions.

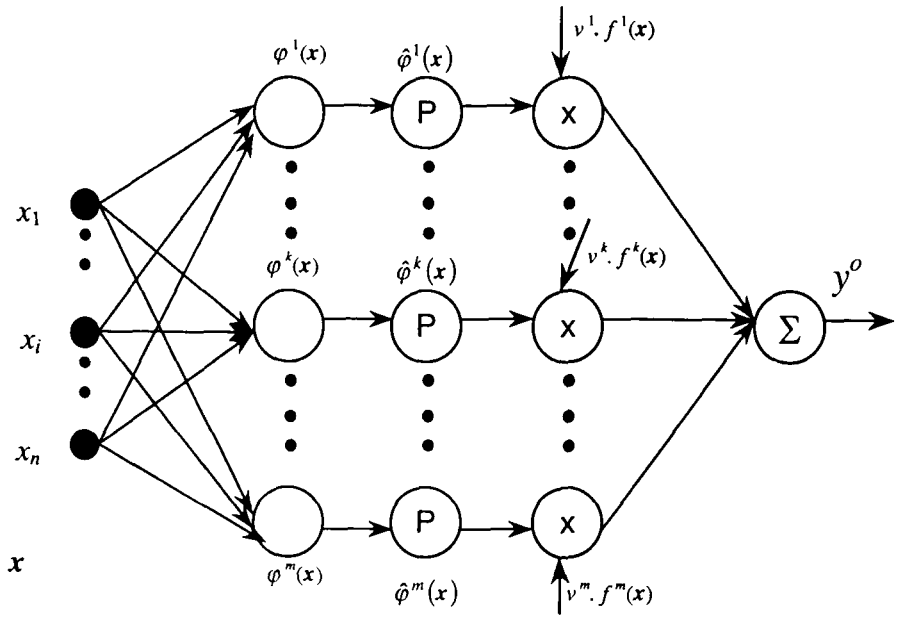


Fig. 3.6: GRBF network for non-normalized calculation.

Among the above listed properties of normalization only "Loss of independence" of basis function poses a difficulty in the learning of normalized network as compared to the learning of non-normalized network. The shape of homogenous function changing to that of non-homogenous function and shift in maxima of basis function do not lead to any side effect, and are sometimes desirable. Covering of the whole input space may result in spurious network output when the input data to the network is not from the learned region of the input space, requiring on-line learning to keep track of the region. Reactivation of basis functions is sometimes advantageous for shifting from one local model to another in the region of overlap as shown in Fig.3.7b for the *Example* of 3.2 below.

Example 3.2:

This example shows the effect of weights v_k on the normalization of basis function $\varphi^k(x)$. Here we deal with the single-input and it can be extended to multi-input case. Using the following two basis functions:

$$\varphi^1(x) = e^{-\left|\frac{x-0.4}{0.1}\right|^{1.5}} \quad , \quad \varphi^2(x) = e^{-\left|\frac{x-0.5}{0.1}\right|^{3.0}}$$

with their local models

$$f^1(x) = 0.6 + 0.4\,x \quad , \quad f^2(x) = 0.7 - 0.4\,x$$

three different cases of v_k are considered

Case 1: $v_1 = v_2$;
Case2: $v_1 = 0.1v_2$;
Case 3: $v_1 = 0.01v_2$;

The basis functions $\varphi^1(x)$ and $\varphi^2(x)$ (solid lines) and corresponding weighted normalized basis functions for case 1 (dashed lines), case 2 (dash-dot lines) and case 3 (dotted lines) are shown in Fig. 3.7a. In Fig. 3.7b the use of reactivation is

demonstrated in the reduction of the number of rules by switching between the two local models in the overlapped region of normalized function.

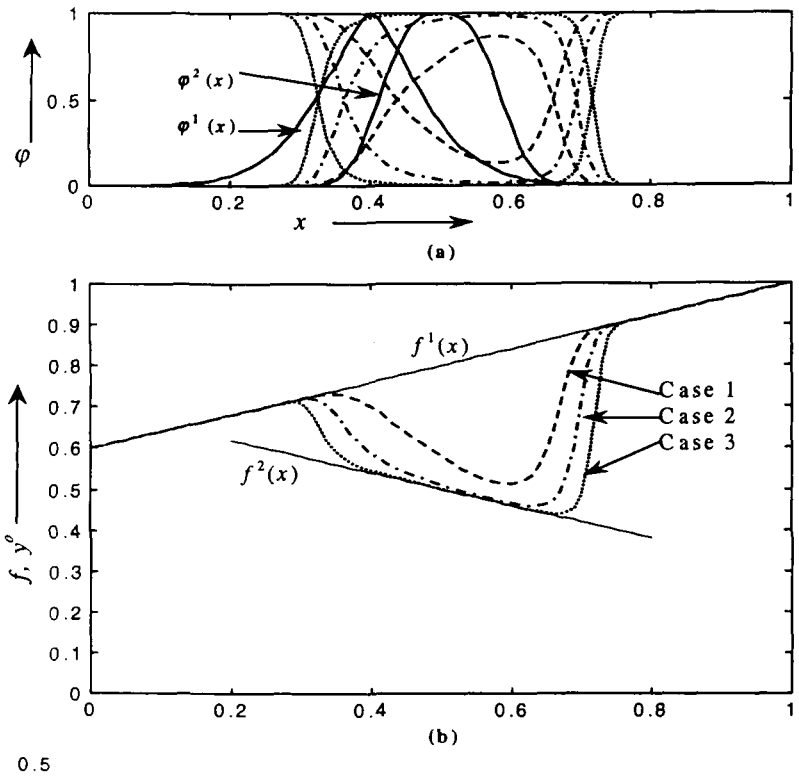


Fig. 3.7: Results of *Example 3.2* (a) effect of v_k on normalization of basis function (b) use of reactivation.

3-4.3 Reduction to the Standard RBF and the Hunt's RBF Networks

We now state the conditions under which the Hunt's network and the standard RBF network are obtained from the generalized form (GRBF network) defined in Sections 3-4.1 and 3-4.2 for both normalized and non-normalized calculations.

1. The function P is unary in layer 2
2. Varying power in each dimension of basis function is capable of handling the insignificant input to that particular basis function (See *Example 3-3*).

3. If each basis function has the same power in each dimension, i.e. $l_{k1} = l_{k2} = \ldots = l_{kn} = 2$, the activation function of each unit in eqn.(3.32) is then simplified to

$$\varphi^k(\boldsymbol{x}) = \exp\left[-\left\{\left(\frac{x_1 - c_{k1}}{\sigma_{k1}}\right)^2 + \left(\frac{x_2 - c_{k2}}{\sigma_{k2}}\right)^2 + \ldots + \left(\frac{x_n - c_{kn}}{\sigma_{kn}}\right)^2\right\}\right] \qquad \ldots(3.37)$$
$$= \exp\left[-(\boldsymbol{x} - \boldsymbol{c}_k)' \Lambda_k (\boldsymbol{x} - \boldsymbol{c}_k)\right]$$

where Λ_k is a width matrix of the form $\Lambda_k = \mathbf{DM}(\boldsymbol{a}_k \text{ .\^{} } 2)$. Equation (3.37) is the generalized form of Gaussian RBF considered in **[Hunt'96]**.

4. In addition to the condition 3, if each basis function has the same width in each dimension, i.e., $\sigma_{k1} = \sigma_{k2} = \ldots = \sigma_{kn} = \sigma_k$, the activation function of each unit in eqn.(3.37) is further simplified to

$$\varphi^k(\boldsymbol{x}) = \exp\left[-\left\{\left(\frac{x_1 - c_{k1}}{\sigma_k}\right)^2 + \left(\frac{x_2 - c_{k2}}{\sigma_k}\right)^2 + \ldots + \left(\frac{x_n - c_{kn}}{\sigma_k}\right)^2\right\}\right] \qquad \ldots(3.38)$$
$$= \exp\left[-\frac{(\boldsymbol{x} - \boldsymbol{c}_k)'(\boldsymbol{x} - \boldsymbol{c}_k)}{\sigma_k^2}\right]$$

which is the standard form of Gaussian RBF considered in **[Jang'93a]**.

5. Fuzziness v_k around each local model $f^k(\boldsymbol{x})$ should be the same. It means that $v_1 = v_2 = \ldots = v_m = v$. Consequently the output of the network is reduced to:

$$y^o = \sum_{k=1}^{m} \frac{\hat{\varphi}^k(\boldsymbol{x})}{\sum_{k=1}^{m} \hat{\varphi}^k(\boldsymbol{x})} . f^k(\boldsymbol{x}) \qquad \ldots(3.39)$$

which is the output of the normalized form of Hunt's RBF network.

6. In addition to the condition 5, if each local model $f^k(\boldsymbol{x})$ is a singleton, i.e., $f^k(\boldsymbol{x}) = b_k$, the output of the network is further simplified to:

$$y^o = \sum_{k=1}^{m} \frac{\hat{\varphi}^k(\boldsymbol{x})}{\sum_{k=1}^{m} \hat{\varphi}^k(\boldsymbol{x})} . b_k \qquad \ldots(3.40)$$

which is the output of normalized form of standard RBF network. For non-normalized network the conditions 1 to 4 and 6 are valid. However, the condition 5 is not required because the outputs of Generalized form of our network and Hunt's network are the same except the basis function.

$$y^o = \sum_{k=1}^{m} \hat{\varphi}^k(\boldsymbol{x}) . v_k . f^k(\boldsymbol{x})$$

$$= \sum_{k=1}^{m} \hat{\varphi}^k(\boldsymbol{x}) . \theta^k(\boldsymbol{x}) \qquad \ldots(3.41)$$

where $\theta^k(\boldsymbol{x}) = v_k . f^k(\boldsymbol{x})$. This is the output of the non-normalized form of Hunt's RBF network.

3-4.4 Symmetry of GRBF networks

Here, we state the symmetric property of GRBF networks and give a logical proof. This symmetric property associated with error surface generated due to approximate solution obtained by GRBF network.

Theorem 3.1: *The GRBF networks (whether in the normalized mode or in the non-normalized mode) with m basis function units yield the m! possible symmetry to the error surface for any unique solution. This can only be*

now searched in $1/m!$ region of the hyperspace of dimension (4n+2), where n is the dimension of the input vector.

Proof: The columns of the network matrix $\boldsymbol{S}$ in (3.36), which represent m basis function parameters (c_k, a_k, l_k) form a total of (4n+2) rows. Out of these, 3n rows correspond to n inputs, one row for the forward link weights (v_k only for the normalized network) and (n+1) rows for the consequent parameters (b_k) of local model $f^k(x)$. Since parameters of each row of $\boldsymbol{S}$ have the same Universe of Discourse (UoD), the columns of $\boldsymbol{S}$ can be interchanged. Column s_1 can be placed in m possible ways. Column s_2 can be placed in the rest of m-1 possible ways, and so on. Column s_{m-1} can be placed in the last 2 possible ways and column s_m can be placed only in 1 way. Hence for the evaluation of any performance of the model, the network matrix $\boldsymbol{S}$ can be written in $m.(m-1).(m-2)...2.1 = m!$ possible ways. Thus, the error surface generated while solving for solution with $\boldsymbol{S}$ will have that $m!$ order of symmetry. The symmetric property helps in the initialization of parameters of GRBF networks.

We now illustrate the symmetry of error surface with a function

We build a model for a given uni-variable function $f(x) = 3x(x-1)(x-1.9)(x+0.7)(x+1.8)$ with two fuzzy rules:

R^1 : if x is A^1 then $y = f^1(x)$;

and R^2 : if x is A^2 then $y = f^2(x)$;

where A^1 and A^2 represent fuzzy sets for the rule R^1 and R^2 respectively. The GRBF nodes which represent A^1 and A^2 have the functions, of the form given by eqn.(3.32), i.e., $\varphi^1(x) = e^{-\left|\frac{x-c_1}{\sigma_1}\right|^{l_1}}$ and $\varphi^2(x) = e^{-\left|\frac{x-c_2}{\sigma_2}\right|^{l_2}}$ respectively.

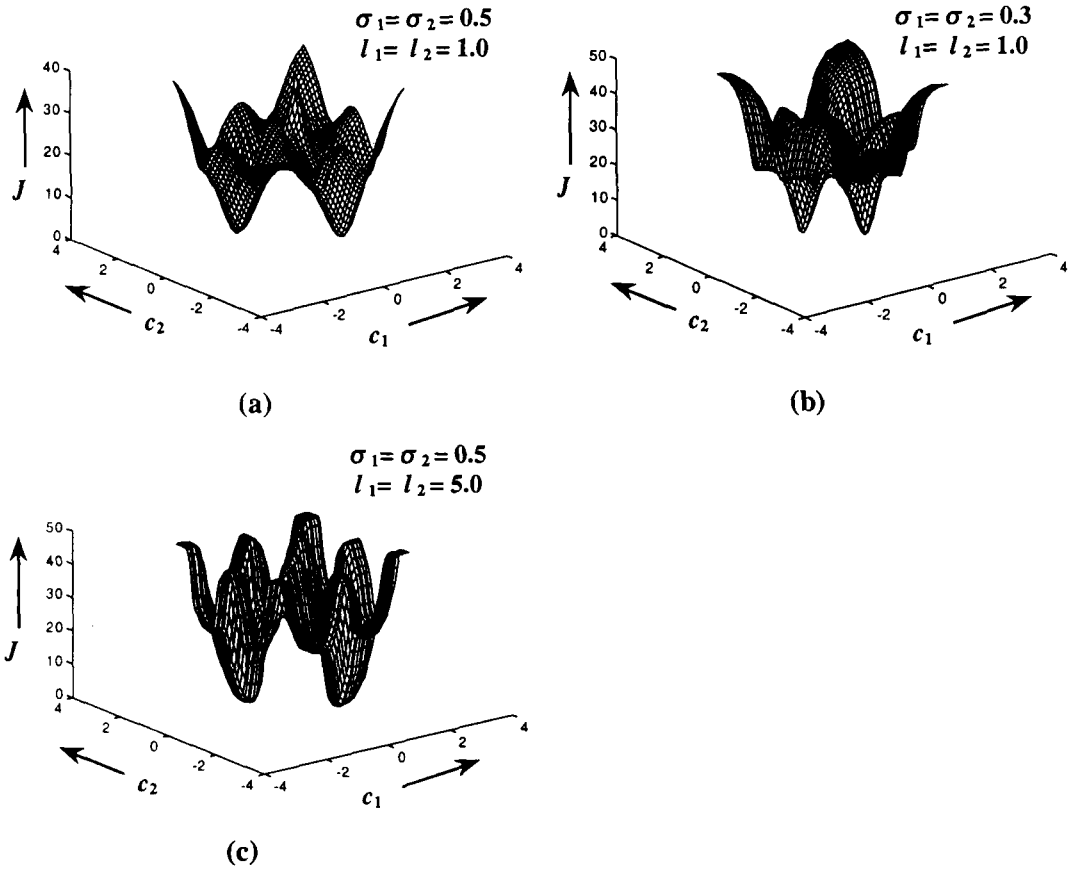


Fig.3.8: The symmetry of the error surface for three sets of values of σ_k's and l_k's over the span of c_k's.

The symmetry of the error surface can be visualized in the Fig.3.8. Figure 3.8(a-c) shows the plots of model performance J (RMS of model error) for three sets of values of σ_k's and l_k's over the span of c_k's, where k= 1, 2. It is observed from the plots of Fig. 3.8(a-c) that the positions of minima and maxima do not change with change in values of σ_k's and l_k's, but only the value for J is different at those

positions. Here the number of rules is 2 so the symmetry of minima and maxima (whether local or global) of error surface is 2! =2, which can be noticed in Fig. 3.8(a-c).

Since the global minimum on the error surface in the parametric hyperspace has the multiplicity of $m!$, we need only consider the $1/m!$ region of parametric hyperspace generated by the above parameters. This fact helps in the proper initialization of the parameters. The symmetry property can be used to generate initialization by partitioning the parametric hyperspace. Much more work needs to be done to come up with a scheme.

3-5 Functional Equivalence

It is shown by Jang-Sun **[Jang'93a]** and Hunt et.al **[Hunt'96]** that Standard RBF network and Hunt's RBF network are functionally equivalent to the TS-model of Fuzzy Inference System (FIS) under the conditions given in the following.

3-5.1 Results of Jang-Sun

1. The number of basis function units is equal to the number of fuzzy if-then rules.
2. The consequent part of each fuzzy rule is a constant.
3. The membership functions of the premise variables within each fuzzy rule are Gaussian functions with the same WGTF.
4. The T-norm operator used to compute the firing strength of each rule is of multiplication type.

5. Both RBFN and the fuzzy inference system under consideration use the same method (i.e., normalized or non-normalized calculation) to derive the overall output.

3-5.2 Results of Hunt et al.

Hunt et al., **[Hunt'96]** have reported that the conditions 1,4 and 5 of Jang-Sun are the same as their conditions 1,3 and 4 for functional equivalence of FIS and RBF. Further they have found that the conditions 2 and 3 of Jang-Sun place a restriction on a particular class of TS-model to which these results apply. To remove these restrictions they have defined a RBF network with local model and discarded the condition 2 and modified the condition 3 of Jang-Sun to condition 2 as stated below:

2. The membership functions for the premise variable within each fuzzy rule are Gaussian functions.

In our proposed framework of GFM, it seems that the conditions 1 and 4 of Jang-Sun, and the conditions 1 and 3 of Hunt et al., are the natural conditions to make the two systems functionally equivalent, without loss of generality, in the type of RBF or FIS. The conditions 2 and 3 of Jang-Sun are the restricted for TS-model with constant consequent and CRI-model with singleton consequent. The condition 2 of Hunt et al., is restricted only to TS-model. The condition 5 of Jang-Sun or condition 4 of Hunt et al., is not required to mention because it is obvious that in both systems we must use the same method to evaluate the overall output. Our aim here is to remove the condition 4 and restriction in the condition 2 from the results of Hunt et al. Above all, we need to add one more condition, for aggregating the output space. The purpose of adding this condition is to incorporate the class I and the class II CRI-models in the functional equivalence of FIS with RBF network.

3-5.3 Generalized Functional Equivalence.

The functional equivalence of the proposed GRBF network with GFM can be established as follows:

Theorem 3.2: *The GRBF network defined by eqns.(3.23-3.36) is functionally equivalent to GFM defined by eqns.(3.19 and 3.20) if the following conditions are satisfied:*

1. *The number of basis function units is equal to the number of fuzzy if-then rules.*
2. *The membership functions for the premise variable within each fuzzy rule are convex and in closed form.[here we use functions as defined by (3.21)]*
3. *The T-norm operator, used to compute the firing strength of each rule, is of multiplication type.*
4. *The S-norm operator, used to join all the mapped region of the output space, is of addition type.*

Proof: Under the conditions 2 and 3, the firing strength of each rule of GFM when $n_k = n$ {which can be met by the definition of premise variable membership function in eqn.(3.21)}, determined from eqn.(3.22)

$$\mu^k(x^k) = \prod_{i=1}^{n} \exp\left(-\left| \frac{x_i - c_{ki}}{\sigma_{ki}} \right|^{l_{ki}} \right) = \exp\left(-\sum_{i=1}^{n} \left| \frac{x_i - c_{ki}}{\sigma_{ki}} \right|^{l_{ki}} \right)$$

and the output of each basis function node of layer 1 determined from eqn.(3.32)

$$\varphi^k(x) = \exp\left[-\left\{ \left| \frac{x_1 - c_{k1}}{\sigma_{k1}} \right|^{l_{k1}} + \left| \frac{x_2 - c_{k2}}{\sigma_{k2}} \right|^{l_{k2}} + \ldots + \left| \frac{x_n - c_{kn}}{\sigma_{kn}} \right|^{l_{kn}} \right\} \right]$$

are identical. Under the condition 1, the overall output of class II GFM evaluated under the condition 4 and given by eqn.(3.20), is identical to the output of GRBF network given by eqn.(3.26). For the unary nodes in layer 2 of GRBF network, the network output is identical to the output of class I GFM given by eqn.(3.19).

Example 3.3:

We consider the similar example as in **[Hunt'96]** to illustrate how the basis function defined by eqn.(3.21) takes care of insignificant premise variables. This example is of TS-model with two premise variables denoted by $\boldsymbol{x} = \{x_1, x_2\}'$ and one output y. The model consisting of five rules.

$$R^1 : \text{if } x_1 \text{ is } \boldsymbol{A} \wedge x_2 \text{ is } \boldsymbol{D} \text{ then } y \text{ is } f^1(\boldsymbol{x})$$

$$R^2 : \text{if } x_1 \text{ is } \boldsymbol{B} \text{ then } y \text{ is } f^2(\boldsymbol{x})$$

$$R^3 : \text{if } x_1 \text{ is } \boldsymbol{C} \wedge x_2 \text{ is } \boldsymbol{E} \text{ then } y \text{ is } f^3(\boldsymbol{x})$$

$$R^4 : \text{if } x_1 \text{ is } \boldsymbol{C} \wedge x_2 \text{ is } \boldsymbol{F} \text{ then } y \text{ is } f^4(\boldsymbol{x})$$

$$R^5 : \text{if } x_2 \text{ is } \boldsymbol{G} \text{ then } y \text{ is } f^5(\boldsymbol{x})$$

The membership functions ***A-G*** are shown in Fig.3.9. We can utilize the property of varying shapes of membership functions given by eqn.(3.21) to define dummy membership functions $\boldsymbol{D_d}$ for x_2 in R^2 and $\boldsymbol{C_d}$ for x_1 in R^5 as shown in Fig.3.9. Then the fuzzy rules R^1-R^5 can be re written as:

$$R^1 : \text{if } x_1 \text{ is } \boldsymbol{A} \wedge x_2 \text{ is } \boldsymbol{D} \text{ then } y \text{ is } f^1(\boldsymbol{x})$$

$$R^2 : \text{if } x_1 \text{ is } \boldsymbol{B} \wedge x_2 \text{ is } \boldsymbol{D_d} \text{ then } y \text{ is } f^2(\boldsymbol{x})$$

$$R^3 : \text{if } x_1 \text{ is } \boldsymbol{C} \wedge x_2 \text{ is } \boldsymbol{E} \text{ then } y \text{ is } f^3(\boldsymbol{x})$$

$$R^4 : \text{if } x_1 \text{ is } \boldsymbol{C} \wedge x_2 \text{ is } \boldsymbol{F} \text{ then } y \text{ is } f^4(\boldsymbol{x})$$

$$R^5 : \text{if } x_1 \text{ is } \boldsymbol{C_d} \wedge x_2 \text{ is } \boldsymbol{G} \text{ then } y \text{ is } f^5(\boldsymbol{x})$$

Where, the premise variable membership functions as defined by the functional forms in Section 3-3.1. are A(15,0.18,2), B(60, 0.14, 3) C(35, 0.11, 2.5), C_d(**50, 0.01, 5**), D(40, 0.12, 3), D_d(**50, 0.01, 5**), E(20, 0.16, 2.5), F(65, 0.13, 2.5), G(85, 0.16, 1.5). Using the parameters of the functional forms, the parameter matrix S for GRBF network can be written as:

$$S = \begin{bmatrix} 15 & 60 & 35 & 35 & \mathbf{50} \\ 40 & \mathbf{50} & 20 & 65 & 85 \\ 0.18 & 0.14 & 0.11 & 0.11 & \mathbf{0.01} \\ 0.12 & \mathbf{0.01} & 0.16 & 0.13 & 0.16 \\ 2 & 3 & 2.5 & 2.5 & \mathbf{5} \\ 3 & \mathbf{5} & 2.5 & 2.5 & 1.5 \\ f^1 & f^2 & f^3 & f^4 & f^5 \\ v_1 & v_2 & v_3 & v_4 & v_5 \end{bmatrix}$$

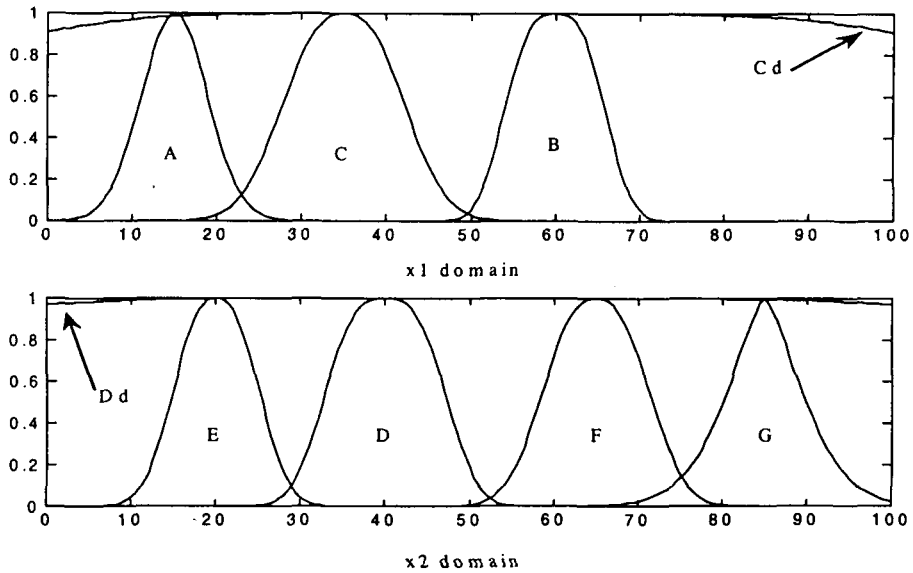


Fig. 3.9. Membership Function for *Example 3.3*

All the RBF unit activation functions corresponding to each fuzzy rule are shown in Fig.3.10. It can be seen that the output of unit 2 is independent of value of x_2 and the output of unit 5 is independent of value of x_1.

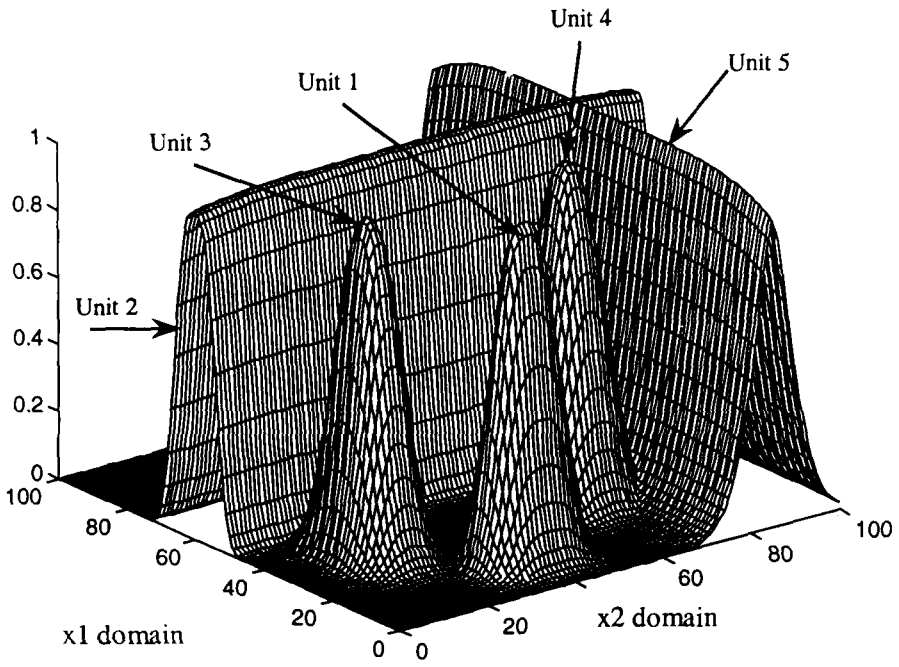


Fig. 3.10. GRB Functions ---- unit 1-5 for example 3.3

Example 3.4:

This example is considered in support of the use of normalized network over non-normalized network and to show how can we remove a rule from a set of fuzzy rules. In this example it is also shown how a fuzzy rule can be eliminated and a premise variable from a rule can be removed in a learned model. Let us consider a non-linear function with two premise variables: $\boldsymbol{x} = \{x_1, x_2\}'$ and one output y. The model consists of six rules. Consider the following function for generating the output data

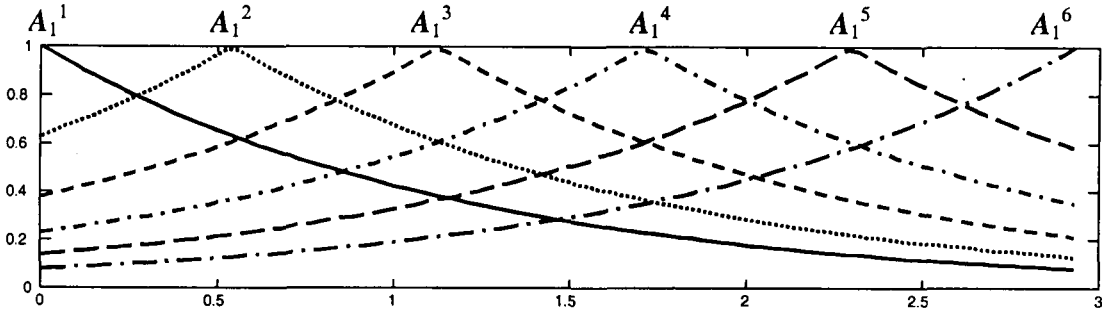
$$y = \left(2 + x_1^{1.5} + 1.5 \sin 3x_2\right)^2; \qquad 0 \leq x_1, x_2 \leq 3$$

where $\boldsymbol{x}$ is an uncorrelated random sequence. Here, we have two cases corresponding to the non-normalized and the normalized networks. In both cases, fuzzy curve is used for the initialization of network and the initial fuzzy rules are listed below.

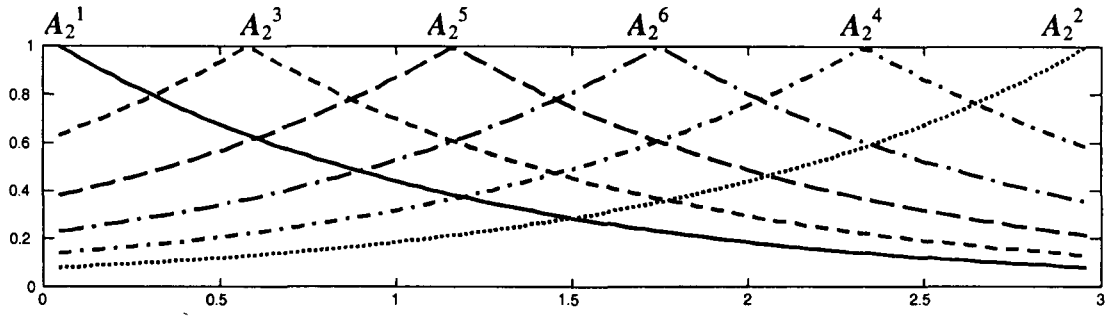
Initialization

R^1 : if x_1 is A_1^1 $\wedge x_2$ is A_2^1 then y is $B^1(0.32,10.28)$;
R^2 : if x_1 is A_1^2 $\wedge x_2$ is A_2^2 then y is $B^2(10.60,10.28)$;
R^3 : if x_1 is A_1^3 $\wedge x_2$ is A_2^3 then y is $B^3(20.88,10.28)$;
R^4 : if x_1 is A_1^4 $\wedge x_2$ is A_2^4 then y is $B^4(31.17,10.28)$;
R^5 : if x_1 is A_1^5 $\wedge x_2$ is A_2^5 then y is $B^5(41.45,10.28)$;
R^6 : if x_1 is A_1^6 $\wedge x_2$ is A_2^6 then y is $B^6(51.73,10.28)$;

The premise variable membership functions A_1^1 to A_1^6 and A_2^1 to A_2^6 are shown in Figs. 3.11a & 3.11b respectively.



(a) x_1 domain



(b) x_2 domain

Fig. 3.11. Initial Premise variable membership functions of *Example 3.4*

We have used only GD method for learning the GRBF network since the consequent part has a fuzzy membership function with centroid and area, which represents CRI-model. The fuzzy rules corresponding to the learned non-normalized network are listed below:

The learned premise variable membership functions A_1^{1f} to A_1^{6f} and A_2^{1f} to A_2^{6f} are shown in Figs.3.12a & 3.12b. In Fig.3.12b, A_2^{2f} the membership function of x_2 remains very small over the domain of x_2, so the fuzzy rule R^{2f} may be removed from the rule base and further learning may be carried out if the performance the of model is not appropriate after removing the rule R^{2f}.

R^{1f} : if x_1 is A_1^{1f} $\wedge$ x_2 is A_2^{1f} then y is $B^{1f}(-0.03,10.28)$;

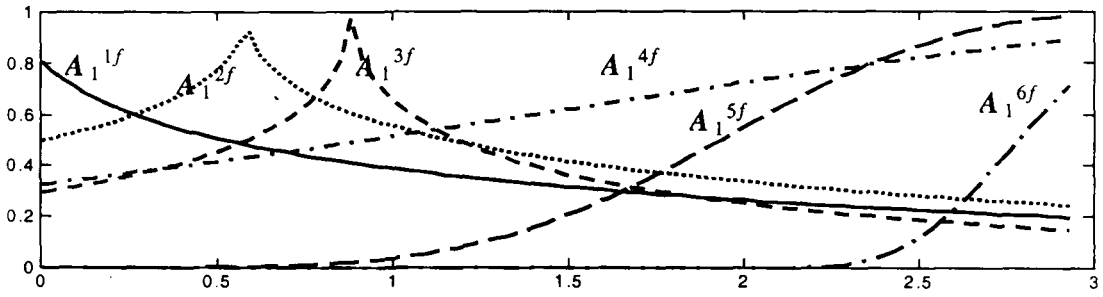
~~R^{2f} : if x_1 is A_1^{2f} $\wedge$ x_2 is A_2^{2f} then y is $B^{2f}(10.32,10.28)$;~~

R^{3f} : if x_1 is A_1^{3f} $\wedge$ x_2 is A_2^{3f} then y is $B^{3f}(20.09,10.28)$;

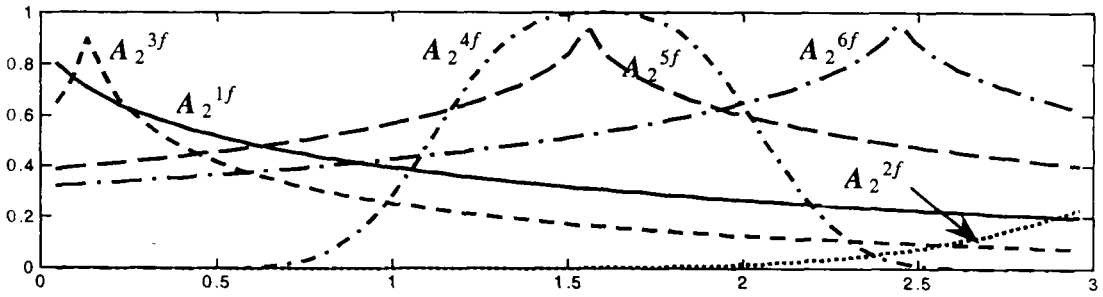
R^{4f} : if x_1 is A_1^{4f} $\wedge$ x_2 is A_2^{4f} then y is $B^{4f}(30.41,10.28)$;

R^{5f} : if x_1 is A_1^{5f} $\wedge$ x_2 is A_2^{5f} then y is $B^{5f}(41.42,10.28)$;

R^{6f} : if x_1 is A_1^{6f} $\wedge$ x_2 is A_2^{6f} then y is $B^{6f}(50.47,10.28)$;



(a) x_1 domain



(b) x2 domain

Fig. 3.12. Final premise variable membership functions for non- normalized network of *Example 3.4*

The fuzzy rules corresponding to the learned normalized network are listed below:

R_n^{1f} : if x_1 is A_{1n}^{1f} $\wedge$ x_2 is A_{2n}^{1f} then y is $B_n^{1f}(-0.58, 10.60)$;

R_n^{2f} : if x_1 is A_{1n}^{2f} $\wedge$ x_2 is A_{2n}^{2f} then y is $B_n^{2f}(9.68,10.00)$;

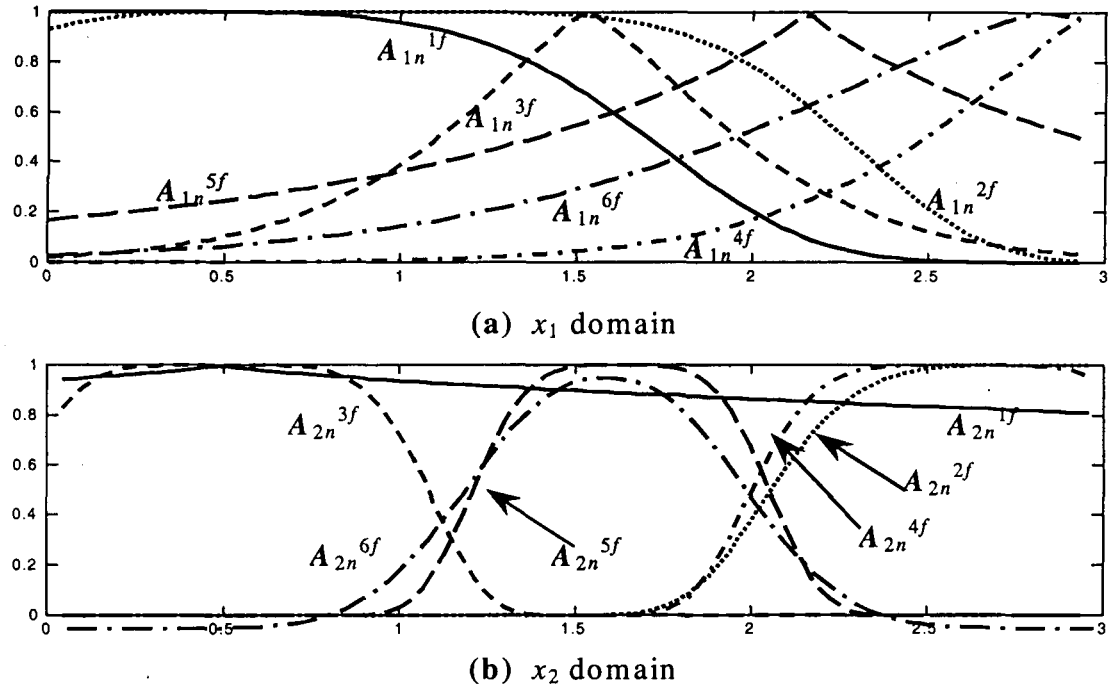
R_n^{3f} : if x_1 is A_{1n}^{3f} $\wedge$ x_2 is A_{2n}^{3f} then y is $B_n^{3f}(20.12,9.53)$;

R_n^{4f} : if x_1 is A_{1n}^{4f} $\wedge x_2$ is A_{2n}^{4f} then y is $B_n^{4f}(30.42,9.78)$;

R_n^{5f} : if x_1 is A_{1n}^{5f} $\wedge x_2$ is A_{2n}^{5f} then y is $B_n^{5f}(40.70,10.15)$;

R_n^{6f} : if x_1 is A_{1n}^{6f} $\wedge x_2$ is A_{2n}^{6f} then y is $B_n^{6f}(50.57,10.09)$;

The learned premise variable membership functions A_{1n}^{1f} to A_{1n}^{6f} and A_{2n}^{1f} to A_{2n}^{6f} are shown in Fig.3.13a & 3.13b.



(a) x_1 domain

(b) x_2 domain

Fig. 3.13: Final Premise variable membership functions for normalized network of *Example 3.4*

In Fig.3.13b, A_{2n}^{1f} membership function of x_2, remains nearly '1' over the domain of x_2, so the variable x_2 may be removed from the fuzzy rule R_n^{1f} without affecting the performance of the fuzzy model and the new fuzzy rules can be listed as in the following:

R_n^{1f} : if x_1 is A_{1n}^{1f} then y is $B_n^{1f}(-0.58, 10.60)$;

R_n^{2f} : if x_1 is A_{1n}^{2f} $\wedge x_2$ is A_{2n}^{2f} then y is $B_n^{2f}(9.68,10.00)$;

R_n^{3f} : if x_1 is A_{1n}^{3f} $\wedge x_2$ is A_{2n}^{3f} then y is $B_n^{3f}(20.12,9.53)$;

R_n^{4f} : if x_1 is A_{1n}^{4f} $\wedge x_2$ is A_{2n}^{4f} then y is $B_n^{4f}(30.42,9.78)$;

R_n^{5f} : if x_1 is A_{1n}^{5f} $\wedge x_2$ is A_{2n}^{5f} then y is $B_n^{5f}(40.70,10.15)$;

R_n^{6f} : if x_1 is A_{1n}^{6f} $\wedge x_2$ is A_{2n}^{6f} then y is $B_n^{6f}(50.57,10.09)$;

Figure 3.14 shows the learning pattern of the non-normalized network (dotted line) and the normalized network (solid line). It can be seen from Fig.3.14 that the performance of the normalized network is better than that of the non-normalized network.

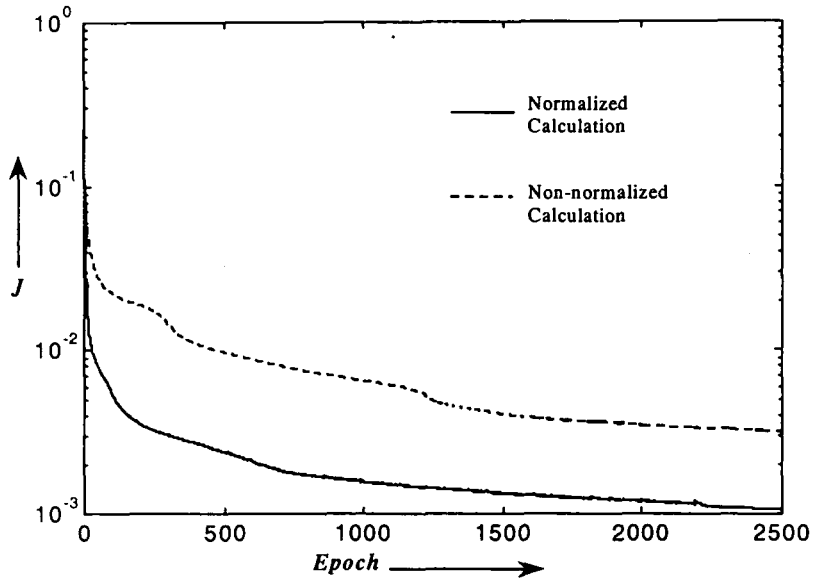


Fig.3.14: Learning pattern of normalized model (—) & non-normalized model (- - -) for *Example 3.4*

In brief, the generalized membership function for the premise variable as defined in eqn.(3.21) takes care of insignificant premise variables in a fuzzy if-then rule and removes the unnecessary fuzzy rules from the rule base as shown by examples 3.3 and 3.4. It also provides a uniform basis function description for all the basis function nodes of RBF network. This leads to easiness in handling the parameters at each node during the learning of RBF network, as compared to the learning of RBF network of Hunt et.al. Better performance of the normalized network over the non-normalized network is also achieved in Example 3.4.

3-6 Simulation Results

In this Section we present the simulation results of the comparison of learning patterns and the final performance of different models. The data of *Example2.1* to *Example2.4*, discussed in chapter 2, are taken for the simulation. The inputs are the same, as selected for the respective examples in Section 2-5, in building models. The fuzzy curve method of structure identification presented in the next chapter (See Section 4-3) is used to obtain the minimum number of rules m_{min} and the corresponding initial parameters of fuzzy sets.

A hybrid of Least Square Estimate (LSE) method and Gradient Descent (GD) learning is applied for fine-tuning of parameters presented in the chapter 5 (See Section 5-2). GD technique updates the model premise parameters and the consequent parameters, i.e., index of fuzziness v_k, while, LSE evaluates the parameters of $f^k(\boldsymbol{x})$ for a given set of premise parameters and consequent parameter v_k. For fine-tuning of models, the algorithm is run for 300 epochs without checking convergence.

For *TS*-model parameter v_k and for Hunt's model parameters v_k and l_k are not updated. Since *CRI*-model has a constant term for $f^k(\boldsymbol{x})$, GD method alone performs the fine-tuning of premise and consequent parameters. One additional model, in which premise parameters are not updated, is considered to show the learning ability of GFM with v_k.

Figures 3.15(a-e) show the learning patterns of different models for *Example2.1* to *Example2.4*. The final values of models for *Example2.1* to *Example2.4* are listed in Table 3-1. In Fig.3.15(a-b) learning patterns of the class I and class II GFM (with v_k and $f^k(\boldsymbol{x})$ only) are not shown because they don't show converging pattern. In Fig.3.15d an excellent ability of learning for this model is shown.

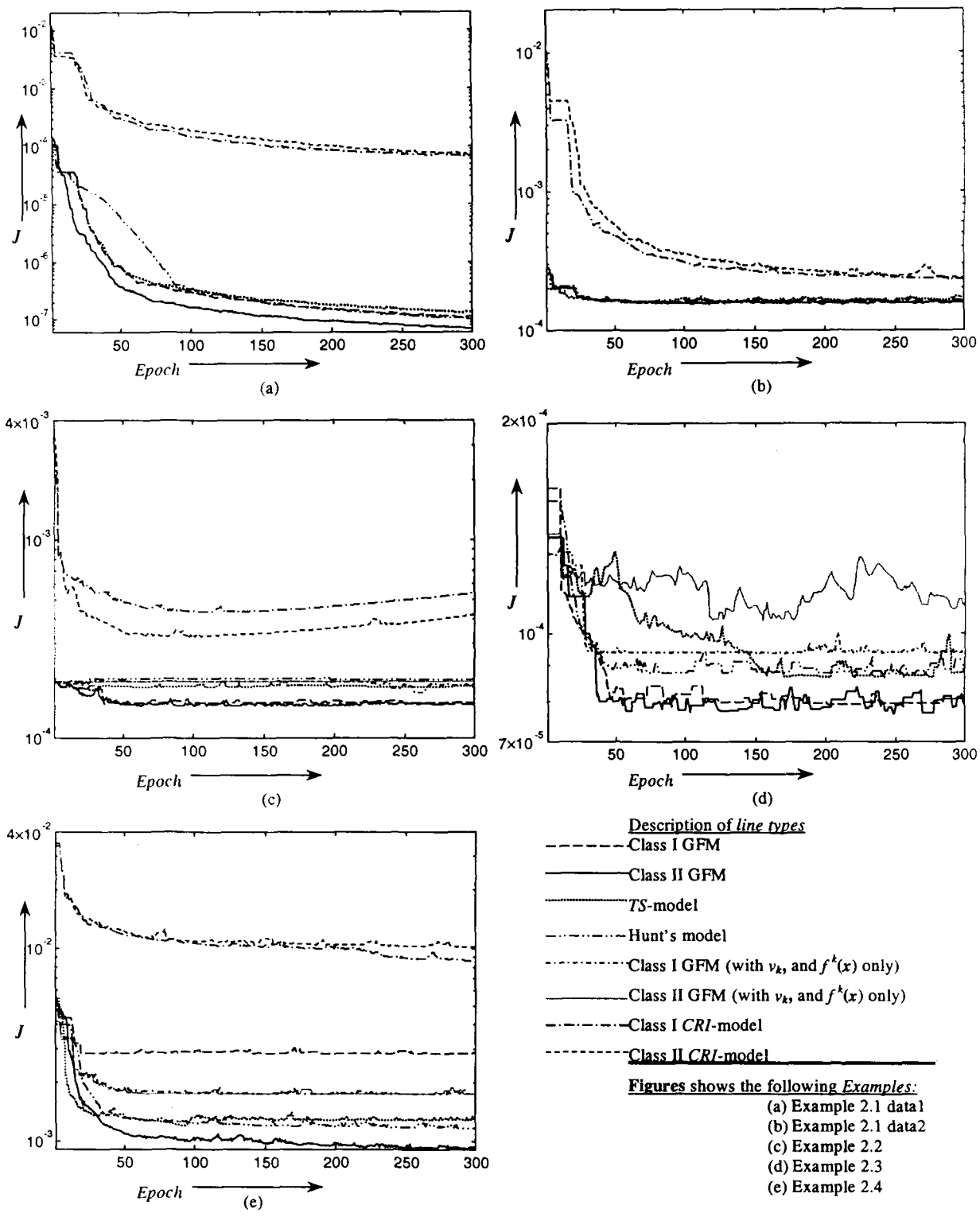


Fig.3.15: Comparative study of learning pattern of different models for *Example2.1-Example2.4.*

The rise in the learning pattern is due to the momentum present in GD technique. The class II GFM has the lowest value of J in all the examples as shown in Fig.3.15(a-e) and given in Table 3-1.

TABLE 3-1: List of Model performance for *Example2.1* to *Example2.4*.

S. No.	*Examples*→ / Cases of ↓ Models		*Example 2.1*		*Ex. 2.2* (Gas Furnace)	*Ex. 2.3* (Human operation at a chemical plant)	*Ex. 2.4* (Daily Stock price)
			Noise free data1	Noisy data2			
1.	GFM	Class I	1.09×10^{-7}	1.56×10^{-4}	1.49×10^{-4}	7.93×10^{-5}	2.8×10^{-3}
		Class II	**6.85×10^{-8}**	**1.53×10^{-4}**	**1.46×10^{-4}**	**7.67×10^{-5}**	**9.11×10^{-4}**
2.	*TS*-Model		1.29×10^{-7}	1.57×10^{-4}	1.76×10^{-4}	8.66×10^{-5}	1.2×10^{-3}
3.	Hunt's Model		1.02×10^{-7}	1.56×10^{-4}	1.86×10^{-4}	8.78×10^{-5}	1.2×10^{-3}
4.	GFM with v_k, and $f^k(x)$ only	Class I	1.26×10^{-4}	2.84×10^{-4}	1.69×10^{-4}	9.37×10^{-5}	1.8×10^{-3}
		Class II	8.78×10^{-5}	2.41×10^{-4}	1.88×10^{-4}	1.04×10^{-4}	1.8×10^{-3}
5.	*CRI*-Model	Class I	6.71×10^{-5}	2.32×10^{-4}	4.34×10^{-4}	0.0057	8.6×10^{-3}
		Class II	7.17×10^{-5}	2.37×10^{-4}	3.26×10^{-4}	0.0028	1.01×10^{-2}

3-7 Conclusions

It is shown that TS-model and CRI-model are special cases of the proposed GFM under certain conditions. By using the generalized basis functions and normalization weights that represent the validity of the local model, a GRBF network is proposed. The functional equivalence of GFM and GRBF is established thus paving the way for the learning ability of GFM. The conditions under which this network reduces to the standard RBF and the Hunt's RBF are also derived. The issue of normalization of basis function is discussed and some new findings are brought out.

The symmetric property of GRBF network is logically proved from the parameter matrix of the GRBF network. The offshoot of this has an important bearing on the hyperspace of error surface. It is demonstrated through examples how an unnecessary rule from a rule base of the learned models can be eliminated and an insignificant variable from a learned rule can be removed using the parameters of learned GRBF network. The main contributions of this chapter are:

- Formulation of GFM
- Architecture of GRBF to learn the parameters of GFM
- Functional equivalence of GRBF with GFM
- Symmetric property of GRBF
- Comparison of model performance

The symmetry of error surface arising out of parameter matrix has a ramification in the choice of appropriate initialization, as we need only to consider $1/m!$ region of the parametric hyperspace.

Chapter 4

STRUCTURE IDENTIFICATION OF GFM

4-1 Introduction

It has been shown in **[Powell'90; Light'92]** that RBF networks can approximate any multivariate function if a sufficient number of RBF units is given. The performance of a RBF network critically depends upon the number as well as position of chosen centers. Different approaches **[Chen'91; Wang'92; Tan'94; Billings'95; Yengwie'96; Lengari'97]** are adopted for: (i) learning the width and centers of RBF units and (ii) learning the connection weights from hidden layers to the output layer of RBF networks. In these learning algorithms, the network structure, or the number of RBF units is predetermined. An appropriate network structure can only be determined by trial and error. In **[Chen'91]** and **[Wang'92]** the network was trained using an orthogonal least square (OLS) algorithm. Akaike's Information Criterion (*AIC*) is used to determine the number of RBF units and an error reduction ratio is used to select the centers **[Akaike'74]**. The OLS algorithm evaluates RBF values for all the data sets, instead of getting centers in hyperspace. Hence OLS algorithm is essentially a block data algorithm. Billings and Zheng **[Billings'95]** have used a modified form of *AIC* by using Mean Square Error (MSE) instead of error variance to determine the number of RBF nodes and center positions which are found by the Genetic Algorithm (GA). GA has its own limitation of time in searching for a

global minimum. In this work *AIC* is used to make a compromise between the model performance and its complexity. Since the first term of AIC, which is a representation of the performance of a model, is heavily weighted by the number of training data sets, so *AIC* is not suitable for determining the number of RBF units.

The organization of this chapter is as follows: The computational complexity of gradient descent learning is established in Section 4-2 in terms of number of RBF nodes in addition to proposing Structure Identification Criterion (*SIC*), independent of the number of training data sets. Fuzzy curve based structure identification which includes the initialization of parameters of fuzzy model is presented in Section 4-3. Fuzzy C-means clustering in $x \times y$ hyperspace **[Azeem'98b]** and Modified mountain clustering method **[Azeem'99a]**, for the initialization of GRBF network are presented in Section 4-4 and Section 4-5 respectively. Simulation results on different Examples of dynamic systems are discussed in Section 4-6 followed by the conclusions in Section 4-7.

4-2 Computational Complexity

Structure Identification of a fuzzy model consists of determining a suitable number and shape of fuzzy partitioning of input-output space, since the number of fuzzy partitions gives the number of rules and the shape of fuzzy partition determines the membership function parameters. Since rules do not cover the rectangular shapes in the input-output space, the required number of rules is arranged at appropriate positions in the fuzzy space of arbitrary shape. In the present work, GFM is identified using GRBF network based on gradient descent learning. Learning of GRBF requires a suitable number of Ist layer RBF nodes (See Fig.3.5.) and proper initialization of the RBF parameters. The RBF parameters are related to the shape of fuzzy partition and

the number of fuzzy partitions represents the number of Ist layer RBF nodes. A brief discussion of approaches for fuzzy partitioning of the input-output space is presented in the subsequent Section. Using the observed data, there can be a number of structures possible for a fuzzy model. For a fixed number of premise variables with suitable initial fuzzy partitioning parameters, the performance of model may improve at the expense of larger computational time resulting from an increase in GRBF network complexity, as the number of rules in a fuzzy model is increased. Hence a criterion for the verification of a structure is of crucial importance.

Takagi **[Takagi'85]** has suggested a structure while minimizing the mean square error, which requires a maximum of np test models for a maximum of 2^p partitions and n number of premise variables. Sugeno **[Sugeno'88]** presented an unbiased criterion based on the cross-evaluation of two fuzzy models using two subsets of observed data set, uniformly distributed in the space, for the same structure. The methods do not address the problem of computational complexity.

With a view to fit a statistical model, Akaike **[Akaike'74]** has suggested the information theoretic criterion *AIC* on the assumption that residuals obey a normal distribution. The *AIC* that makes a compromise between the performance and the number of rules **[Chen'91; Wang92]** is of the form

$$AIC(\chi) = M \ln\left(\sigma_e^2\right) + \chi . m \qquad ...(4.1)$$

where M is the number of observed data points, σ_e^2 is the variance of residual errors and χ is the critical value of the chi-squared distribution with one degree of freedom and for a given significance level. A suitable choice of $\chi = 4$ **[Leontaritis'87]** corresponds to the significance level of 0.0456. In **[Billings'95]** the variance of residual errors is replaced by the mean square error. The first term of the *AIC* highly

depends upon the number of observed data. For large number of observed data, the second term becomes negligible.

To establish a relationship between the number of fuzzy rules and computational complexity, we employ the gradient descent algorithm using the given observed data set for 30 epochs with increasing number of rules. A plot of average time/epoch (T_{av}) vs number of rules is shown in Fig.4.1.(a-d) on different scales. It is clear from Fig.4.1.(a-d) that the computational complexity is of exponential nature $\mathcal{O}\left(e^m\right)$. This form is obvious from the figures, but, the mathematical justification is yet to be found.

To provide a compromise between the model performance and computational complexity we minimize the Structure Identification Criterion (*SIC*). *SIC* defined as the natural logarithm of product of the Mean Square Error (MSE) and the Computational Complexity (CC) giving equal weight to MSE and CC, is:

$$\begin{aligned} SIC &= \ln\left(J.e^m\right) \\ &= \ln\left(J\right)+m \end{aligned} \qquad \ldots(4.2)$$

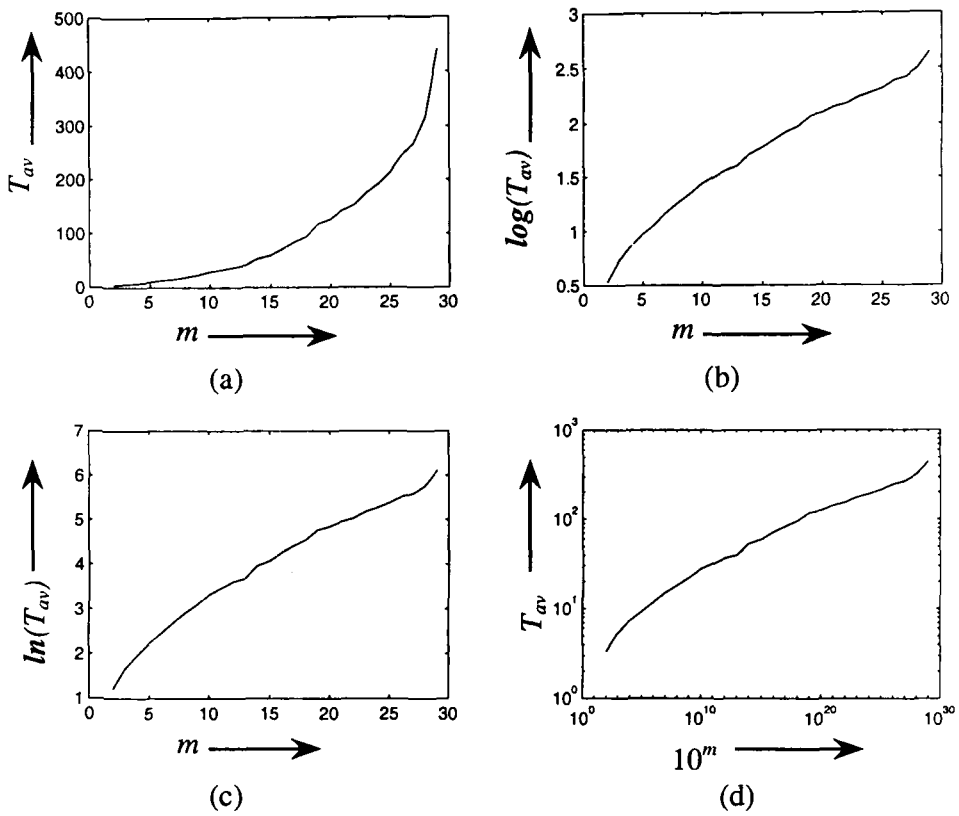


Fig. 4.1:. Plots of (a) Average time/epoch (T_{av}) vs number of rules on linear scale
(b) Logarithmic of average time/epoch (T_{av}) vs number of rules on linear scale
(c) Natural logarithmic of average time/epoch (T_{av}) vs number of rules on linear scale.
(d) Average time/epoch (T_{av}) vs decimal exponential of number of rules on logarithmic scale.

4-3 Fuzzy curve based Structure Identification

We make use of fuzzy curve method of chapter 2 for the determination of the number of rules based on a heuristic. We then specify the membership functions and then initialize the parameters of the membership functions being trained by a GRBF network.

4-3.1 Heuristic for the Number of rules

Determination of the number of rules is a very crucial problem in the design of GFM. Here we shall explore the use of fuzzy curve in determining the number of

rules. The maxima and minima of fuzzy curve give the minimum number of rules needed to approximate each fuzzy curve y_i^o. This is a heuristic devised on the concept that fuzzy model will interpolate between the maxima and minima. If the maxima and minima are far apart, or the curve is not smooth, a rule may be added. For n inputs we have n fuzzy curves, so we have n different numbers $m_1, m_2, \ldots, m_n$. The minimum number of rules m_{min}, is $m_{min} = \mathbf{max}(m_1, m_2, ..., m_n)$ and the maximum number of rules m_{max}, is $m_{max} = (m_1 \times m_2 \times ... \times m_n)$ needed in GRBF network implementation of GFM.

4-3.2 Fuzzy partitioning of premise and consequent Variables

For the proposed architecture of GRBF network, the number of fuzzy partitions of all the variables is equal to the number of rules determined by the above procedure. Initially the middle point of the first and the last fuzzy partitions are at the beginning and at the end of the range of each variable and the middle points of the rest of the fuzzy partitions (m - 2 in number) are located at the interval of {range/(m-1)}. We take an adaptive membership function for each input partition and triangular membership function for each output partition. The width is taken as {α×interval}; where $\alpha \in [0.5, 2]$.

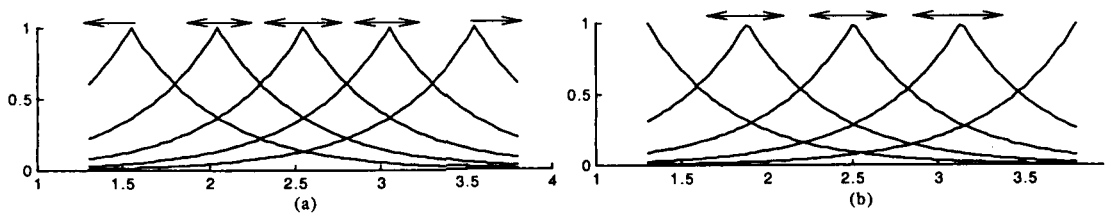


Fig. 4.2: Initialization of membership functions (a) Lin method (b) Proposed method

Initialization of fuzzy membership functions using the approach in **[Lin'95]** and that using the proposed approach are shown in Fig.4.2(a) and Fig. 4.2 (b) respectively. Convergence of GRBF network requires the least number of epochs by the proposed approach than in **[Lin'95]**. During the training of GRBF network, central values of the first and the last fuzzy subsets (for which the membership value is 1) move towards the extreme ends {shown by arrows in Fig.4.2(a)}, in order to cover the entire Universe of Discourse(UoD). On the other hand this is achieved at the time of initialization itself by the proposed approach shown in Fig.4.2(b), thus saving the additional effort..

4-3.3 Initialization of parameters

Parameters b_k , v_k ($k = 1, \ldots, m$) are set to the centers of fuzzy membership functions of a consequent variable. To do this, the following steps are being used:

1. Divide the range of the desired output data into m intervals, such that the width of each interval is equal to range of $y/(m\text{-}1)$, except the widths of I^{st} and the last interval both of which are half of the range of $y/(m\text{-}1)$.
2. Set b_k to the central value of k^{th} interval ($k = 2, 3, \ldots, m\text{-}1$), and set it to the lower end ($k = 1$) and the upper end ($k = m$) of the ranges of the desired output data.
3. Set all v_k equal to range of $y/(m\text{-}1)$.

Next, we make use of fuzzy curves to set the initial parameters c_{ik} and a_{ik} ($i = 1, \ldots, n$) ; ($k = 1, \ldots, m$)} so as to initialize the membership functions for premise variables by following the steps as under:

1. Divide the domain of each fuzzy curve y_i^o into m intervals corresponding to m intervals in the output space, such that the width of each interval is

equal to range of x_i /(m-1). Except the widths of Ist and last interval both of which are half of the range of x_i /(m-1).

2. For the curve y_i^o, label the centers of intervals x_{ik} (k = 2, . . . , {k-1}), for k = 1 the center value is the beginning of the range and for $k = m$ the center value is the end of the range . Order x_{ik} (k = 1, 2, . . . , m) by the corresponding value of y_i^o, then x_{im} corresponds to the interval containing the largest value of y_i^o.
3. The interval containing the point x_{im} is associated with the output interval whose center is at b_m.
4. In a similar way $x_{i(m-1)}$is associated with the center $c_{i(m-1)}$ of interval which contains the next largest central point on the curve y_i^o , and $c_{i(m-1)}$ is associated with b_{m-1}, and so on for $k = m$-2, m-3, . . ., 1.
5. The length of an interval over which a rule is applied in the domain of y_i^o is denoted by Δx_i . Initial fuzzy membership function of x_i for rule k is defined as $\mu_i^k(x_i) = \exp\left(-\left| \frac{x_i - x_{ik}}{\alpha.\Delta x_i} \right|^{l_{ik}} \right)$ where α is typically in the range [0.5, 2]. The initial parameters are $a_{ik} = \frac{1}{\alpha.\Delta x_i}$ and $c_{ik} = x_{ik}$.
6. Set l_{ik} =1or 2, {(i = 1, . . . , n) ; (k = 1, . . . , m)}.

4-4 Fuzzy C-means Clustering in $x \times y$ space

Fuzzy C-means clustering aims at organizing and revealing structures in data sets. Clustering is commonly viewed as an unsupervised learning. Our objective is to

form an appropriate number of clusters, which are the representatives of rule patches, in the normalized $x \times y$ hyperspace known as hypercube. This helps to find the centroid of the consequent part of CRI-model constrained by the output data.

Define the j^{th} data in $x \times y$ hyperspace as follows:

$$x_{jy} \equiv \{x_j, y_j\} = \{x_1(j), x_2(j), \ldots, x_n(j), y(j)\} \ \forall \ \ j = 1, \ldots, M \qquad \ldots(4.3)$$

where, $x_j = \{x_1(j), x_2(j), \ldots, x_n(j)\}$. Since clustering is associated with a geometrical property, without loss of generality, we normalize each dimension of hyperspace by shifting and scaling operation so as to bound the data points in hyper-cube. The normalized data points $\bar{x}_{jp}$ are defined as:

$$\bar{x}_{jy} \equiv \left\langle x_{jy} - (x_{jy})_{\min} \right\rangle / \left\langle (x_{jy})_{\max} - (x_{jy})_{\min} \right\rangle \ \forall \ \ j = 1, \ldots, M \qquad \ldots(4.4)$$

where,

$$(x_{jy})_{\min} = \left\{ \min_{j=1}^{M} x_1, \ \min_{j=1}^{M} x_2, \ \ldots, \ \min_{j=1}^{M} x_n, \ \min_{j=1}^{M} y \right\} \qquad \ldots(4.5)$$

and

$$(x_{jy})_{\max} = \left\{ \max_{j=1}^{M} x_1, \ \max_{j=1}^{M} x_2, \ \ldots, \ \max_{j=1}^{M} x_n, \ \max_{j=1}^{M} y \right\} \qquad \ldots(4.6)$$

The objective function J_{cy} for fuzzy clustering in $x \times y$ hyperspace can be specified as the weighted sum of distances between the datum and the corresponding cluster centers. In general it takes the following form:

$$J_{cy} = \sum_{k=1}^{m} \sum_{j=1}^{M} \mu_{jk}^{q} \, d_{jk}^{2} (\bar{x}_{jy}, \bar{c}_{ky}) \qquad \ldots(4.7)$$

where, $\bar{c}_{ky} = \{\bar{c}_k, \bar{b}^k\} = \{\bar{c}_{k1}, \bar{c}_{k2}, \ldots, \bar{c}_{kn}, \bar{b}^k\}$.

$$c_{ky} = \left[\bar{c}_{ky} .* \left\{ (x_{jy})_{\max} - (x_{jy})_{\min} \right\} \right] + (x_{jy})_{\min}$$

$$d_{jk}^2(\bar{x}_{jy}, \bar{c}_{ky}) = (\bar{x}_{jy} - \bar{c}_{ky}) Q_k (\bar{x}_{jy} - \bar{c}_{ky})'$$

where, $\bar{c}_{ky}$ $(k = 1, \ldots, m)$ are the cluster centers and Q_k is a positive definite matrix in $\mathcal{R}^{(n+1)} \times \mathcal{R}^{(n+1)}$ hyperspace, $1 < q < \infty$, while $\mathbf{U} = [\mu_{jk}]$ is a partition matrix which describes the allocation of data to a particular cluster and satisfies the following two conditions:

$$\sum_{k=1}^{m} \mu_{jk} = 1 \qquad \text{for} \quad j = 1, \ldots, M \qquad \ldots(4.8)$$

$$0 < \sum_{l=1}^{M} \mu_{jk} < M \qquad \text{for} \quad k = 1, \ldots, m \qquad \ldots(4.9)$$

The family of all partition matrices will be denoted by $\mathcal{U}$.

The minimization of objective function J_{cy} is done, with respect to cluster centers. That is,

$$\min_{\bar{c}_{1y}, \bar{c}_{2y}, \ldots, \bar{c}_{my}} J_{cy} \qquad \ldots(4.10)$$

subjected to $\mathbf{U} \in \mathcal{U}$.

Solution of the above minimization problem is as follows:

$$\bar{c}_{ky} = \frac{\sum_{j=1}^{M} \mu_{jk}^q \bar{x}_{jy}}{\sum_{j=1}^{M} \mu_{jk}^q} \qquad \ldots(4.11)$$

$$\mu_{jk} = \frac{\left(\frac{1}{d_{jk}^2(\bar{x}_{jy}, \bar{c}_{ky})} \right)^{\frac{1}{(q-1)}}}{\sum_{k=1}^{m} \left(\frac{1}{d_{jk}^2(\bar{x}_{jy}, \bar{c}_{ky})} \right)^{\frac{1}{(q-1)}}} \qquad \ldots(4.12)$$

(Proof of the above solution is given in *Appendix B*)

The number of optimum clusters for the data set $\mathcal{D}_M = \{x_j, y_j\}_{j=1}^{M}$ is decided by the validity function S which is the ratio of compactness to separation, **[Xie'91:]**

$$S = \frac{\sum_{k=1}^{m}\sum_{j=1}^{M} \mu_{jk}^2 \left\| \bar{x}_{jy} - \bar{c}_{ky} \right\|^2}{M \min_{i \neq j} \left\| \bar{c}_{iy} - \bar{c}_{jy} \right\|^2} \quad ; \text{ for each } m = m_{min}, \ldots, m_{max} \qquad ...(4.13)$$

Denoting the optimal candidate at each m by Ω_m the solution to the following minimization problem

$$\min_{m_{min} \leq m \leq m_{max}} \left(\min_{\Omega_m} S \right) \qquad ...(4.14)$$

is ensured to yield the most valid fuzzy clustering of the data set. S has a tendency to decrease eventually when m is very large. So, the values of S are meaningless when m gets close to M. Since in practice the feasible number of clusters m is much smaller than the number of data points M, we apply two heuristic methods for the determination of $m_{\max}$ for both small and large values of M.

In the first method, we plot the optimal values of S for $m = m_{min}$ to M -1 when M takes small values, and select the starting point of monotonically decreasing tendency of S at m_{max}.

In the second method we need not compute S for very large m when M takes large values. It is almost always the case that m at the stop-value is << M. We can choose m_{max} a priori, e.g., say $m_{max} = M/3$ which is not likely to reach the starting point of the decreasing tendency.

4-5 Modified Mountain Clustering

The purpose of clustering is to do natural groupings of large set of data, producing a concise representation of system's behavior. Yager and Filev [9] have proposed a, simple and easy to implement, mountain clustering algorithm for estimating the number and location of cluster centers. Their method is a grid based three-step procedure. In the first step the hyperspace is discretized to obtain a grid with a certain resolution in each dimension as shown in Fig.4.3 for a normalized two dimensional plot. The second step uses the data set to construct the mountain function around all grid points. The third step generates the cluster centers by an iterative destruction of mountain function. Though this method is simple but the computation grows exponentially with the increase of dimension of hyperspace and the number of grid points. In the n dimensional hyperspace with the number of grid lines, g in each dimension, the number of grid points that must be evaluated is g^n.

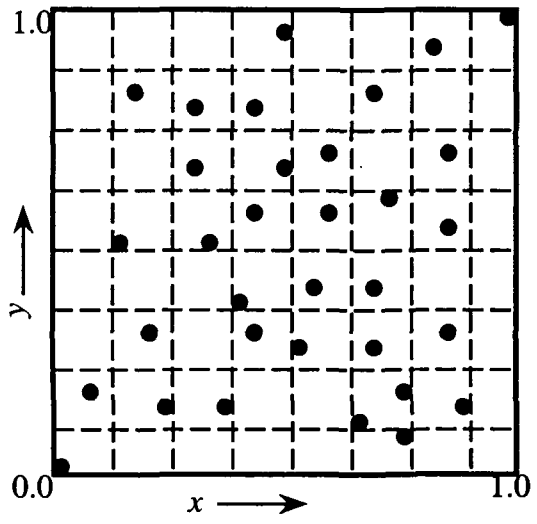


Fig.4.3: Grid points and data points in a normalized two-dimensional plot

We present here a modified form of Yager and Filev's method. We assume that each data point has potential to become a cluster center instead of a grid point.

This modification makes the computation complexity independent of the dimension, because the number of points is equal to the number of data. The second advantage of this modification is that it eliminates the need for a grid, in which case a compromise between the accuracy and the computational complexity must be struck. The procedure of the modified method is as follows:

Treating each data point as a cluster center, we define a measure of potential in eqn.(4.15), which is a mountain function, as a function of distance $d^2(\bar{x}_{ry}, \bar{x}_{jy}) = (\bar{x}_{ry} - \bar{x}_{jy}) Q (\bar{x}_{ry} - \bar{x}_{jy})'$ between $\bar{x}_{ry}$ and all other data points.

$$P_{r,1} = \sum_{j=1}^{M} \exp\left[-\left(\frac{d^2(\bar{x}_{ry}, \bar{x}_{jy})}{d_1^2} \right) \right] \quad \forall r = 1, 2, \ldots, M \qquad \ldots(4.15)$$

Where, Q is a $(n+1)\times(n+1)$ positive definite matrix and d_1 is the positive constant, which defines the neighborhood of datum. Data points outside radial distance d_1 have a little influence on the potential. It is evident from the mountain function that the potential value of datum is an approximation of the density of data point (cardinality) in the vicinity of datum. The higher the potential value of each of the data points in hyper-cube the higher the chance of becoming a cluster center. The first cluster center is selected with the highest value of $P_{r,1}$ as follows:

$$\bar{c}_{1y} = \bar{x}_{1y}^* \Leftarrow P_1^* = \max_{r=1}^{M} (P_{r,1}) \qquad \ldots(4.16)$$

For the selection of second cluster center, the potential value of each datum is revised in order to deduce the effect of mountain function around the first cluster center as follows:

$$P_{r,2} = P_{r,1} - P_1^* \exp\left[-\left(\frac{d^2(\bar{x}_{ry}, \bar{c}_{1y})}{d_2^2} \right) \right] \quad \forall\, r = 1, 2, \ldots, M \qquad \ldots(4.17)$$

where, d_2 is the positive constant, which defines the neighborhood of cluster center. It is evident from eqn.(4.17) that the data points near the first cluster center have greatly reduced potential value and are unlikely to be selected as the next cluster center. After revision of the potential value of each datum, second cluster center is selected with the highest value of $P_{r,2}$ as under:

$$\bar{c}_{2y} = \bar{x}^*_{2y} \Leftarrow P_2^* = \max_{r=1}^{M}(P_{r,2}) \qquad ...(4.18)$$

Similarly, for the selection of k^{th} cluster center, revision of the potential value for each datum is done by

$$P_{r,k} = P_{r,k-1} - P^*_{k-1} \exp\left[-\left(\frac{d^2\left(\bar{x}_{ry}, \bar{c}_{(k-1)y}\right)}{d_2^2}\right)\right] \quad \forall r = 1, 2, \ldots, M \qquad ...(4.19)$$

and k^{th} cluster center is selected with the highest value of $P_{r,k}$ as under

$$\bar{c}_{ky} = \bar{x}^*_{ky} \Leftarrow P_k^* = \max_{r=1}^{M}(P_{r,k}) \qquad ...(4.20)$$

To stop this procedure, Yager and Filev **[Yager'94]** have used the criterion $P_k^* / P_1^* < \delta$ (δ is a small fraction). The choice of δ affects the results. Small value of δ results in a large number of cluster centers vice versa. It is difficult to establish a single value for δ that works well for all data. To overcome this difficulty a gray region of δ value, bounded by limit δ_u and δ_l, is used. The upper limit δ_u is the threshold for absolute acceptance of cluster center and the lower limit δ_l is the threshold for complete rejection and for the end of clustering process. In the gray region, a good trade-off between reasonable potential value and sufficient distance from the existing cluster center is used to accept a datum as a cluster center:

$$\frac{P_k^*}{P_1^*} + \frac{d_{\min}}{d_1} \geq 1 \qquad ...(4.21)$$

where, $d_{\min}$ = minimum distance among $\bar{c}_{ky}$ and the previously selected cluster centers.

The optimum number of clusters for the data set $\mathcal{D}_M = \{x_t, y_t\}_{t=1}^{M}$ is decided by the validity function S defined earlier [See *Appendix B*]

$$S = \frac{\sum_{k=1}^{m}\sum_{r=1}^{M} \bar{\mu}_{r,k}^2 \left\| \bar{x}_{ry} - \bar{c}_{ky} \right\|^2}{M \min_{i \neq j} \left\| \bar{c}_{iy} - \bar{c}_{jy} \right\|^2} \qquad ...(4.22)$$

where, $\bar{\mu}_{r,k} = \mu_{r,k} \Big/ \sum_{k=1}^{m} \mu_{r,k}$. The membership function $\mu_{r,k}$ represents the degree of association of r^{th} data to the k^{th} cluster center and is defined as:

$$\mu_{r,k} = \exp\left[-\left(\frac{d^2\left(\bar{x}_{ry} - \bar{c}_{ky}\right)}{d_2^2} \right) \right] \qquad ...(4.23)$$

Equation (4.23) assures the most valid fuzzy clustering of the data set.

Optimal S candidate corresponds to the parameters $\bar{c}_{ky}^*$ and d_2^*. These parameters are used for the initialization of GFM. The de-normalized value of $\bar{c}_{ky}^*$ is given by

$$c_{ky}^* = \left[\bar{c}_{ky}^* .* \left\{ \left(x_{jy}\right)_{\max} - \left(x_{jy}\right)_{\min} \right\} \right] + \left(x_{jy}\right)_{\min} \qquad ...(4.24)$$

where, $c_{ky}^* = \left\{ c_k, b^k \right\} = \left\{ c_{k1}, c_{k2}, \ldots, c_{kn}, b^k \right\}$.

The initial width of each membership function on each dimension of hyperspace is determined from d_2^* as follows:

$$\sigma_{ky}^* = \left[d_2^* .* \left\{ \left(x_{jy}\right)_{\max} - \left(x_{jy}\right)_{\min} \right\} \right] \qquad ...(4.25)$$

where, $\sigma_{ky}^* = \left\{ \sigma_k, \sigma_y^k \right\} = \left\{ \sigma_{k1}, \sigma_{k2}, \ldots, \sigma_{kn}, v^k \right\}$.

4-5.1 The choice of clustering parameters

The number of resulting cluster centers and the distances among them are highly dependent upon the mountain clustering parameters, i.e., the neighborhood of datum d_1, neighborhood of clusters d_2, gray region of parameter δ_u and δ_l. A brief analysis of their choice is as follows:

(a) Neighborhood of datum d_1

The smaller the value of d_1, smaller the values of P_1^* and d_2 which result in a large number of cluster centers and vice versa. The number of cluster centers approaches the number of data points thereby defeating the purpose of clustering. The maximum value for d_1 is half of the principal diagonal of unit hypercube, i.e., $d_{1,\max} = \sqrt{(n+1)}/2$ and the minimum value may be taken as $d_{1,\min} = 0.2 d_{1,\max}$.

(b) Neighborhood of cluster centers d_2

The spaces among the resulting clusters are highly depend upon the value of d_2. To avoid the resulting cluster centers to be far apart and closely spaced, a good choice is to consider a ratio d_2/d_1 that lies in between 0.75 and 1.0. Since with $d_1 = d_{1,\max}$ the hypersphere inscribes the hypercube, and if $d_2 > d_{1,\max}$ the resulting number of clusters is one that defeats the purpose of fuzzy modeling and is not desirable.

(c) Gray region of parameters δ_u *and* δ_l

The number of cluster centers also depends upon the position and the range of gray region. A good choice of the upper and lower limits is to take δ_u = 0.2 and δ_l=0.02 respectively.

As mentioned above, the number of clusters, which represents the number of rules is decided by validity function, i.e., eqn.(4.14). In other words, we can divide the

input-output space into appropriate sub-spaces that yield the rules. We have assumed an adaptive symmetric membership function around the cluster center projected on each premise variable axis [See eqns.(3.21) and.(3.22)] and a symmetric membership function [See eqn.(3.11)] around the consequent "variable centroid", represented by the local model $f^k(\boldsymbol{x})$, on the consequent variable axis.

4-6 Simulation Results

Fine-tuning of initial GRBF (or fuzzy rules) can be achieved by minimizing the objective function $\boldsymbol{J}$, which is a normalized mean square error [See Section 5-2], with respect to the parameters c_i^k, a_i^k, l_i^k, v^k, and $f^k(\boldsymbol{x})$.

To evaluate the parameters of local model $f^k(\boldsymbol{x})$, GFM is formulated in Least Square Estimate (LSE) framework **[Azeem'99a]**. A hybrid learning algorithm composed of gradient descent learning and LSE learning is used to train GFM equivalent of GRBF network

We present simulation results on four examples of chapter 2. In the *Example 2.1*, *Example 2.2*, and *Example 2.4* the models are learned on 80% of the data and prediction is made for the remaining 20% of the data, while in *Example 2.3* the models have been learned using all the data.

In this section, we consider two classes, i.e., class I and class II GFM models. The rules are obtained for different data sets by applying different structure identification criteria such as S, SIC, AIC. The number of rules for each example is supported by the relevant curves. We also show the errors of modeling using the rules obtained.

The performance of a model improves if the objective function decreases. The number of rules in a fuzzy model is determined by using the measures: ***S, AIC, SIC.*** The plots (for example Figs. 4.4) contain the number of rules on the x-axis and one of the measures on the y-axis. The number of rules, yielding the minimum value of these plots, is selected for a particular measure employed on a particular method. We now describe the results of structure determination on different examples.

Example 2.1 Revisited

The GFM of class I and of class II, obtained by varying the number of rules using the proposed methods on data 1 (i.e., α = 0), are tested for their objective function J, validity measure S, *SIC* and *AIC*. From Figs. 4.4 and 4.5, we observe that the objective function decreases with both fuzzy C-means and modified mountain clustering. In the case of fuzzy curve it reduces and then increases as the number of rules increases and this is due to the improper initialization. For fuzzy C-means and modified mountain clustering we choose the initialization for which the validity measure S is the minimum for a particular structure. On the other hand, with the minimum number of rules, the performances of the models obtained from fuzzy curve and modified mountain clustering are approximately the same. The modified mountain clustering yields a performance that is almost constant beyond 5 rules. As for as S is concerned, again modified mountain clustering yields a slowly varying S as compared to fuzzy C-means. The number of rules is found to be 3 corresponding to the minimum S. The *SIC* criterion operates well on all methods yielding 3 rules, whereas the *AIC* is dependent on the 'number of data'. The number of rules, for the case of fuzzy curve and fuzzy C-means clustering, is found to be 9 that yields the minimum *AIC*, and 3 for the case of mountain clustering. Desired output y (solid line) and model output y^o (dashed line) are shown in Fig.4.6(a) and Fig.4.7(a) for the class

I GFM and class II respectively, obtained from modified mountain clustering method of initialization with 3 rules. The corresponding *model errors* are plotted in Fig.4.6(b) and Fig.4.7(b). The values of J for the class I GFM corresponding to the training and prediction are $3.1377\text{x}10^{-7}$ and $4.8872\text{x}10^{-7}$ and the values of J for the class II GFM are $2.8948\text{x}10^{-7}$ and $4.0606\text{x}10^{-7}$.

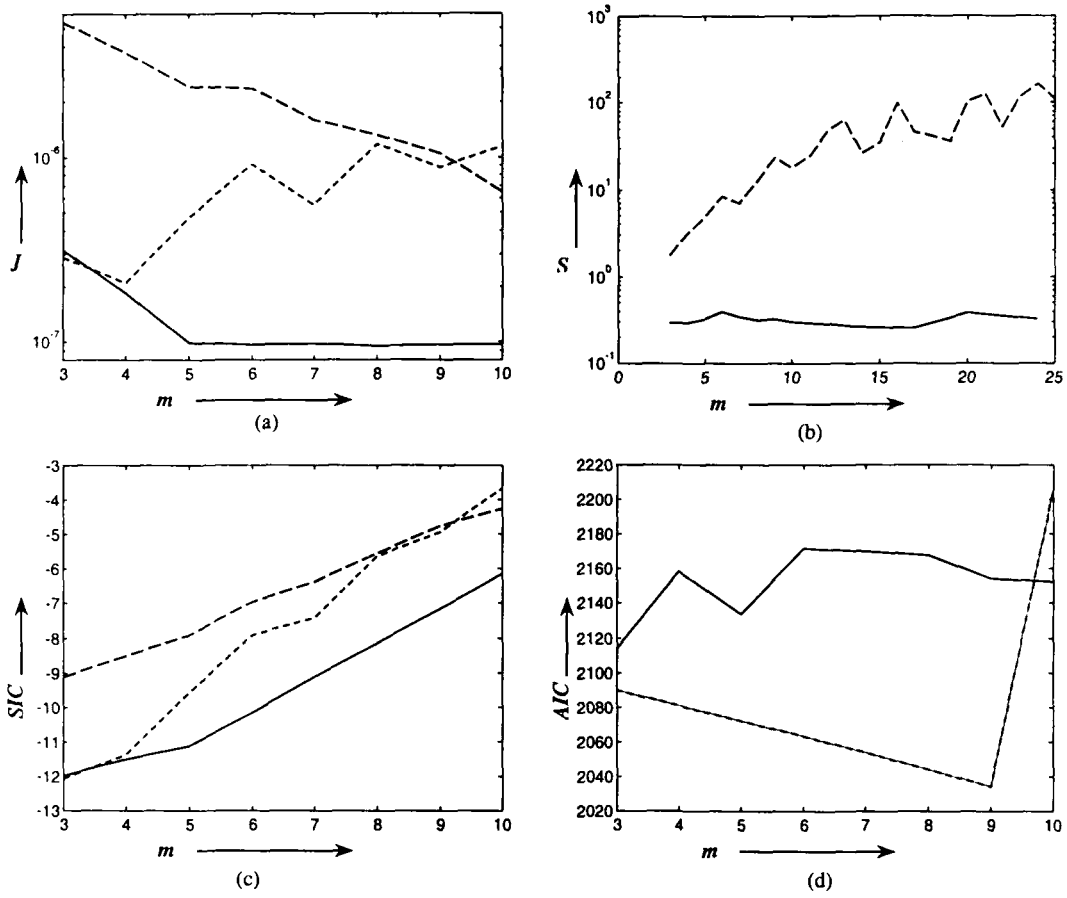


Fig.4.4: Plots of (a) J (b) S (c) ***SIC*** (d) ***AIC*** for class I GFM for noise free data of *Example 2.1* w.r.t. different number of RBF units with different method of initialization of centers, i.e., fuzzy curve (- - - - - - -, short dashed line), Fuzzy C-mean clustering (— — — —, long dashed line) and Modified Mountain clustering (————, solid line).

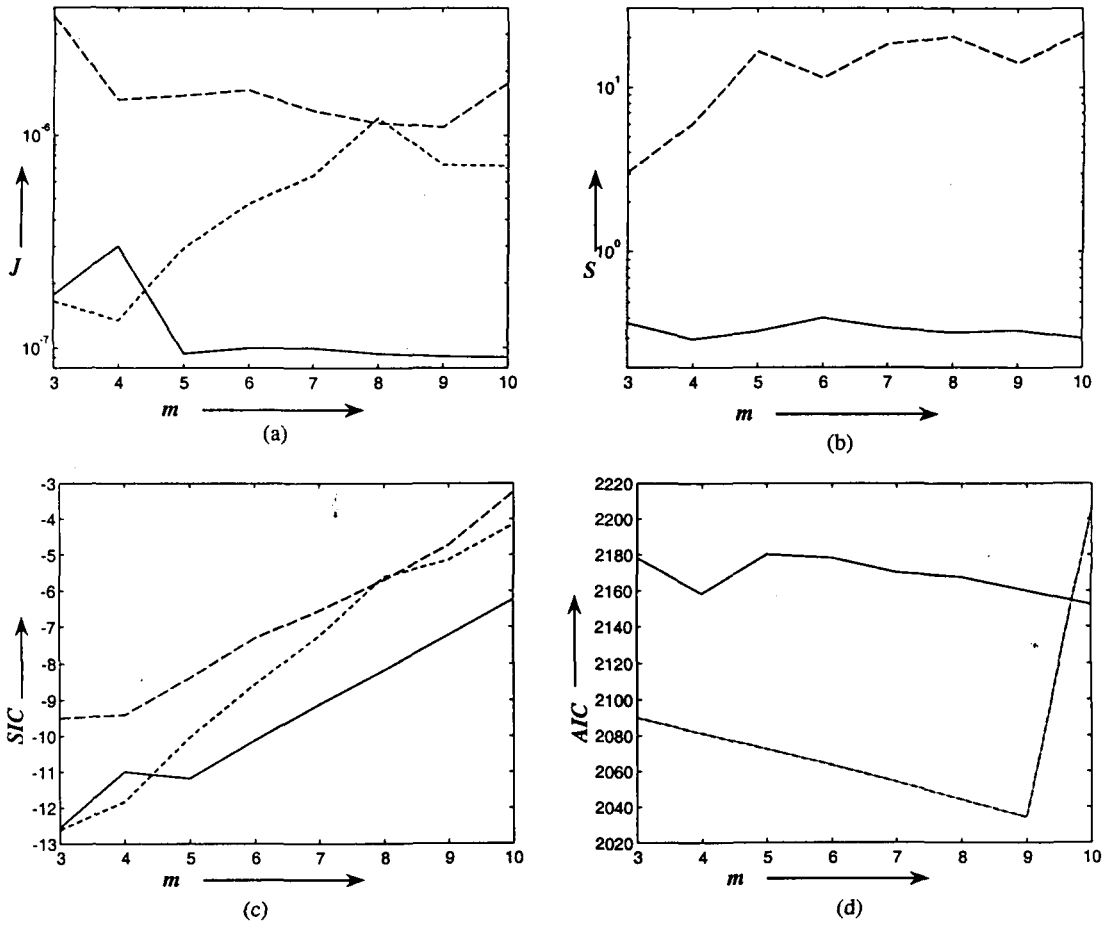


Fig. 4.5: Plots of (a) J (b) S (c) SIC (d) AIC for class II GFM for noise free data of *Example 2.1* w.r.t. different number of RBF units with different method of initialization of centers, i.e., fuzzy curve (- - - - - - -, short dashed line), Fuzzy C-mean clustering (— — — —, long dashed line) and Modified Mountain clustering (————, solid line).

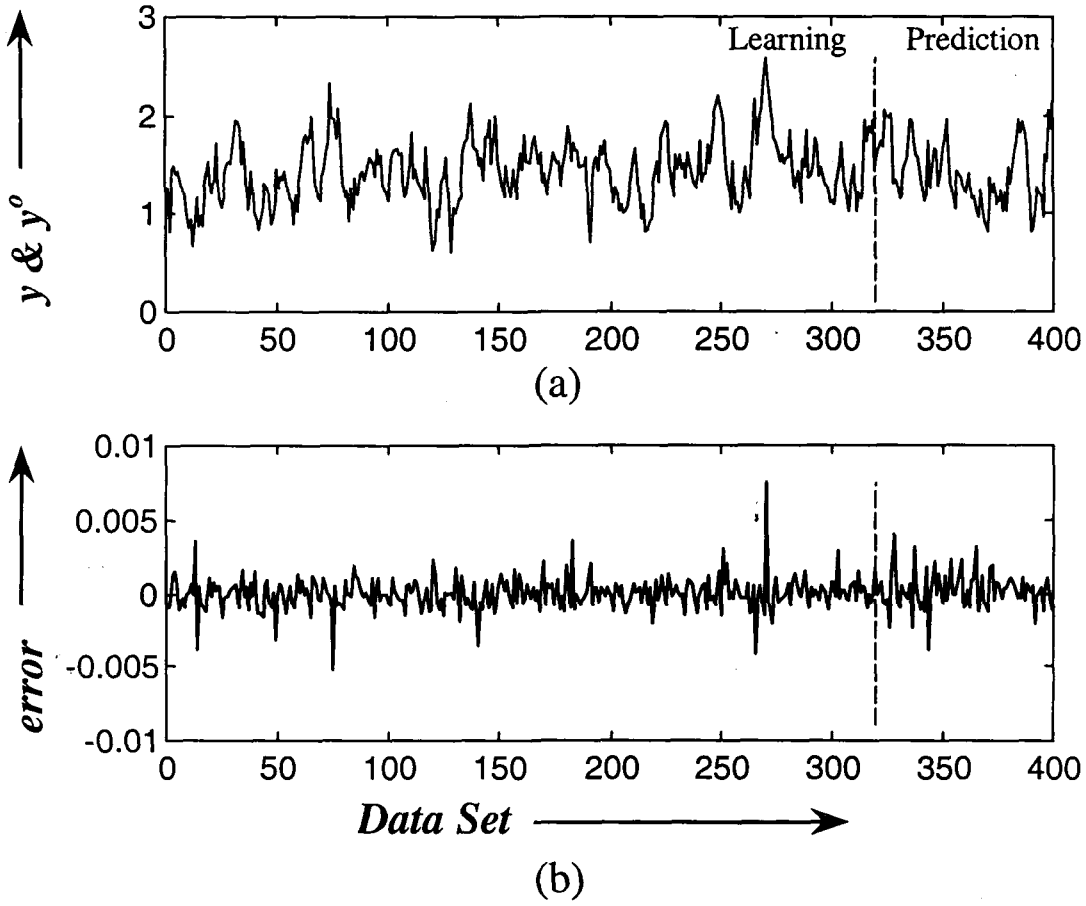


Fig. 4.6: Plots of (a) y (————, solid line) and y^o (- - - - - -,dashed line) (b) *model error* for class I GFM for noise free data of *Example 2.1* obtained from the initialization of centers based on Modified Mountain clustering method with 3 RBF units.

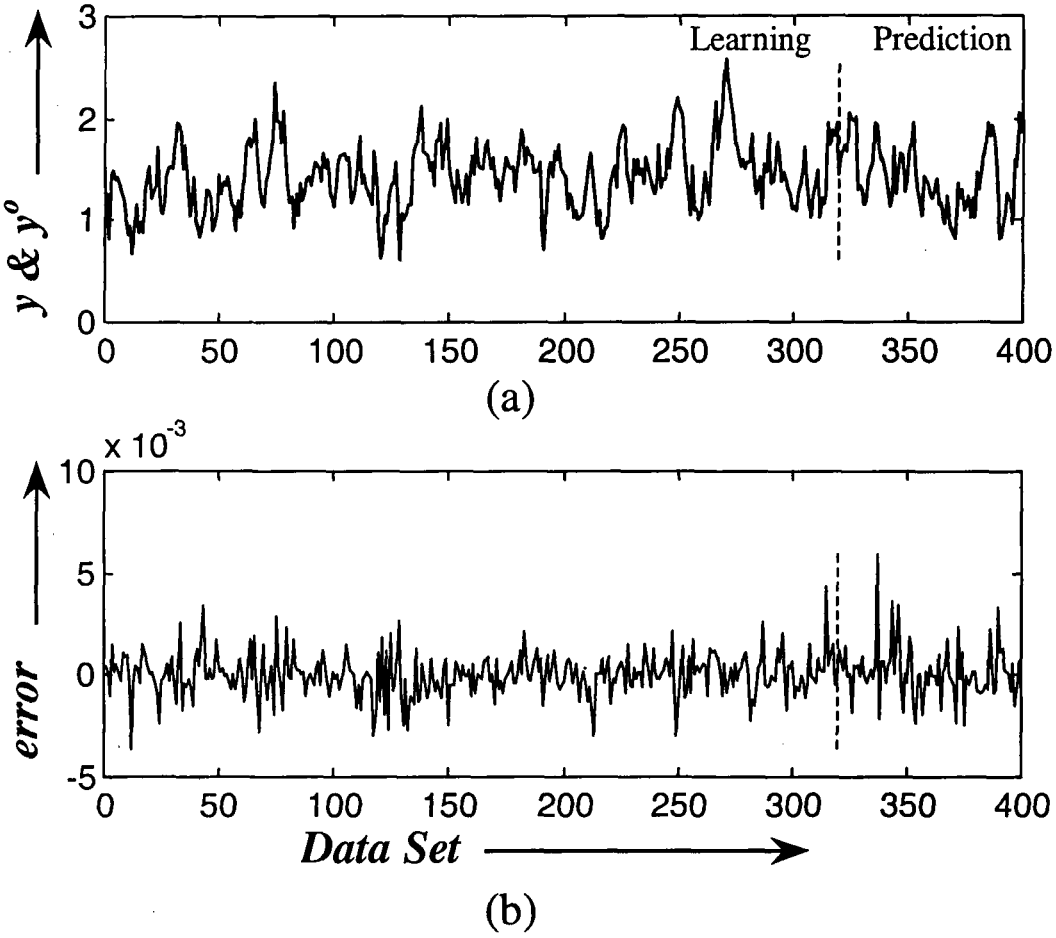


Fig. 4.7: Plots of (a) y (————, solid line) and y^o (- - - - - -,dashed line) (b) *model error* for class II GFM for noise free data of *Example 2.1* obtained from the initialization of centers based on Modified Mountain clustering method with 3 RBF units.

For data 2 (i.e., $\alpha = 1$), we observe from Figs. 4.8 and 4.9 that the objective function decreases as the number of rules is increased, for all the three methods. As for as S is concerned, again modified mountain clustering yields a slowly varying S as compared to fuzzy C-means clustering. The number of rules appears to be 3 corresponding to the minimum S. The *SIC* criterion operates well on all the methods yielding 3 rules. The number of rules, for the case of fuzzy curve and fuzzy C-means clustering, is found to be 9 that yields the minimum *AIC*, and 3 for the case of mountain clustering. Desired output y (solid line) and model output y^o (dashed line) are shown in Fig.10(a) and Fig.11(a) for the class I and class II GFM, respectively, obtained from the modified mountain clustering method of initialization with 3 rules. The corresponding *model errors* are plotted in Fig.10(b) and Fig.11(b). The objective function J of the class I GFM for the training data set is 1.5742×10^{-4} and for prediction is 2.2271×10^{-4}. The objective function J of the class II GFM for the training data set is 1.5054×10^{-4} and for prediction is 1.9604×10^{-4}.

Figures 4.6(b), 4.7(b), 4.10(b) and 4.11(b) show that the learned models predict very well during the prediction period whether the data are noise free (i.e., $\alpha = 1$), or noisy (i.e., $\alpha = 2$).

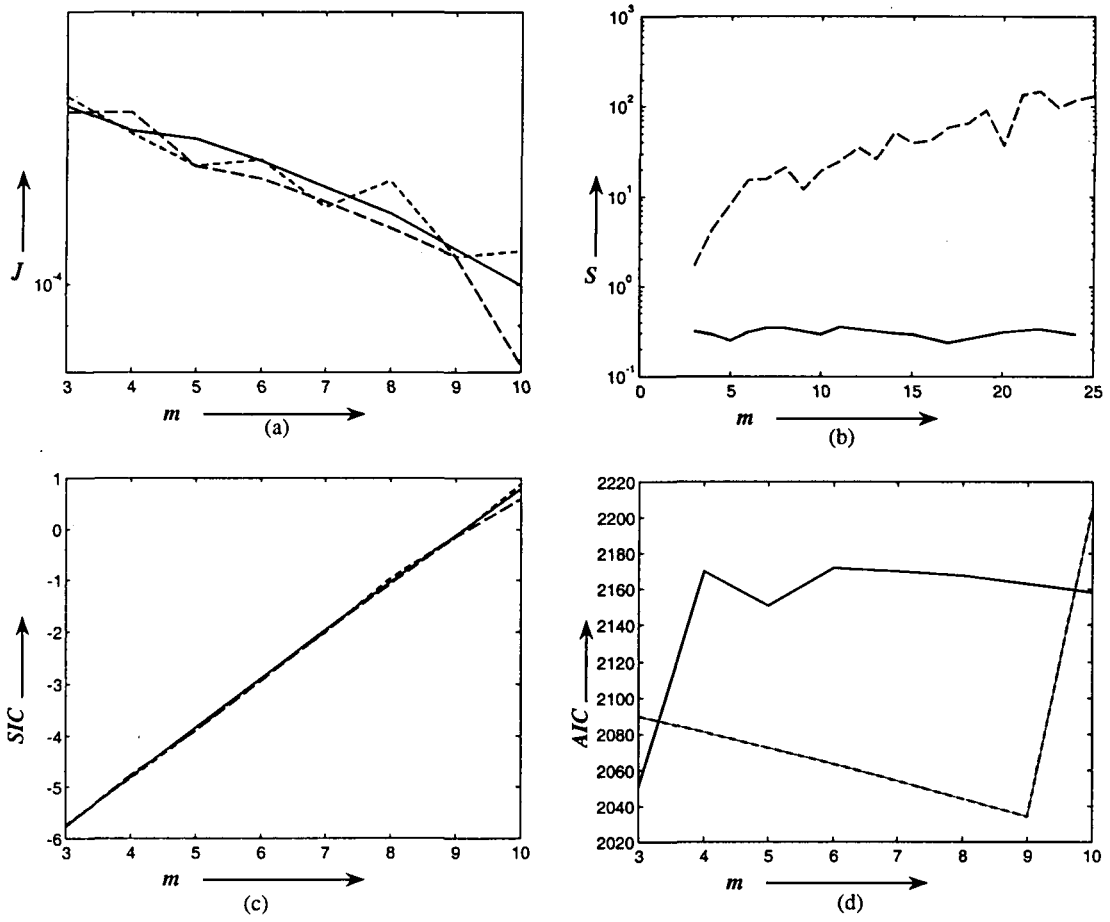


Fig.4.8: Plots of (a) J (b) S (c) ***SIC*** (d) ***AIC*** for class I GFM of noisy data of *Example 2.1* w.r.t. different number of RBF units with different methods of initialization of centers, i.e., fuzzy curve (- - - - - - -, short dashed line), Fuzzy C-means clustering (— — — —, long dashed line) and Modified Mountain clustering (————, solid line).

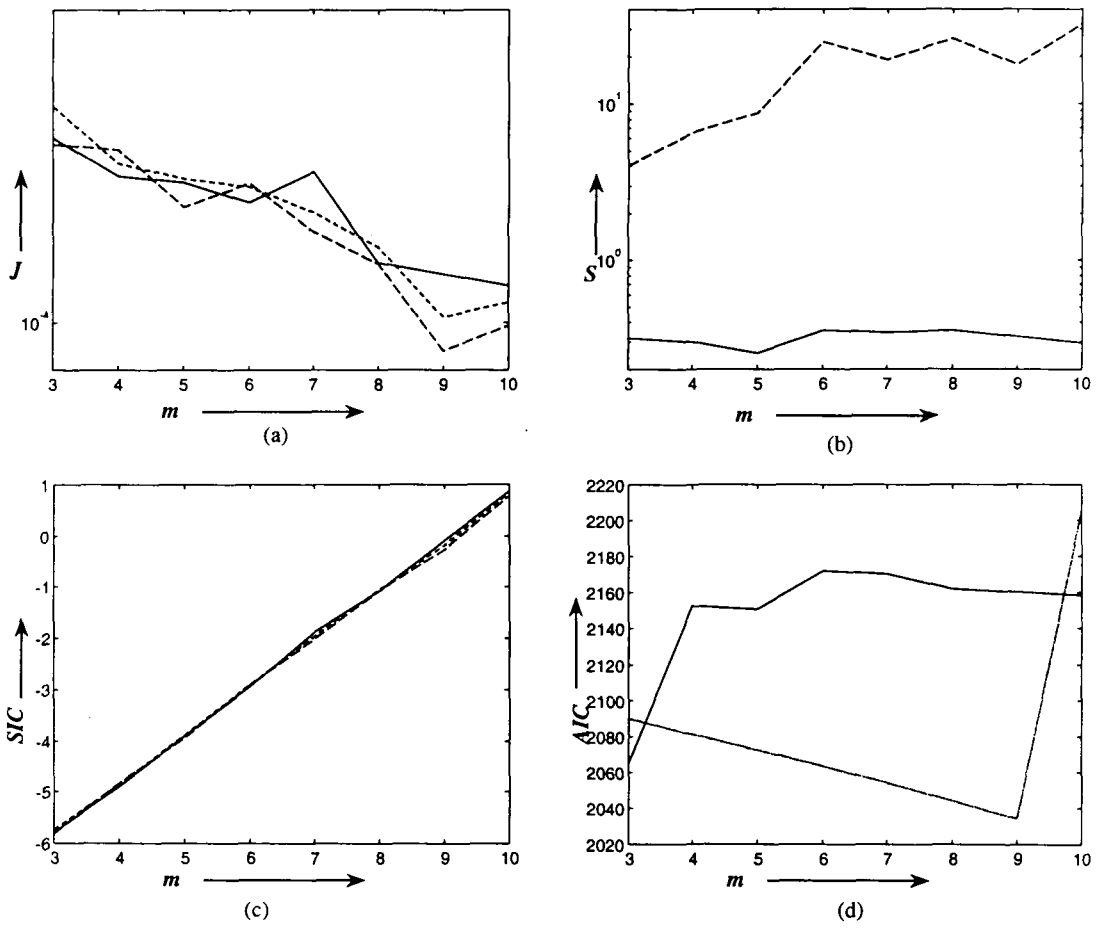


Fig.4.9: Plots of (a) J (b) S (c) SIC (d) AIC for class II GFM of noisy data of *Example 2.1* w.r.t. different number of RBF units with different methods of initialization of centers, i.e., fuzzy curve (- - - - - -, short dashed line), Fuzzy C-means clustering (— — — —, long dashed line) and Modified Mountain clustering (————, solid line).

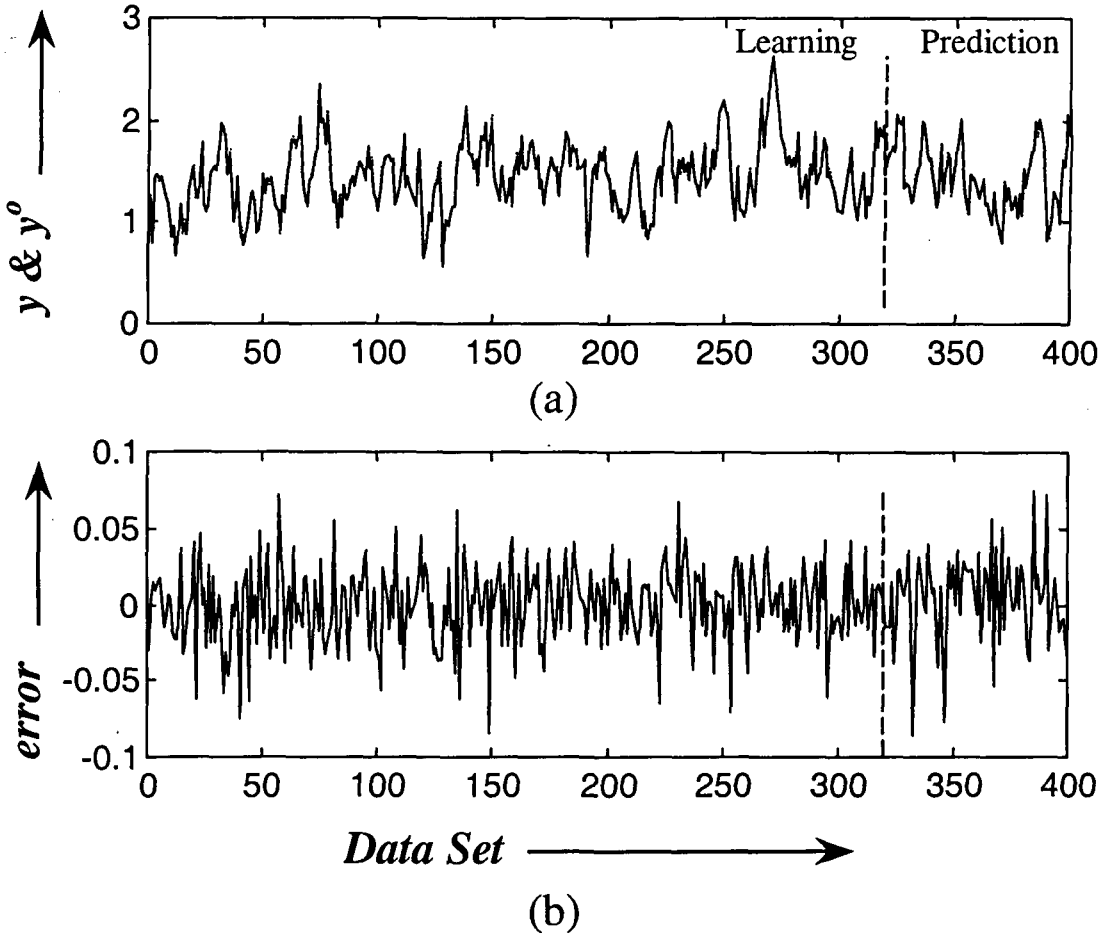


Fig.4.10: Plots of (a) y (————, solid line) and y^o (- - - - - - -,dashed line) (b) *model error* for class I GFM of *Example 2.1* noisy data obtained from the initialization of centers based on Modified Mountain clustering method with 3 RBF units.

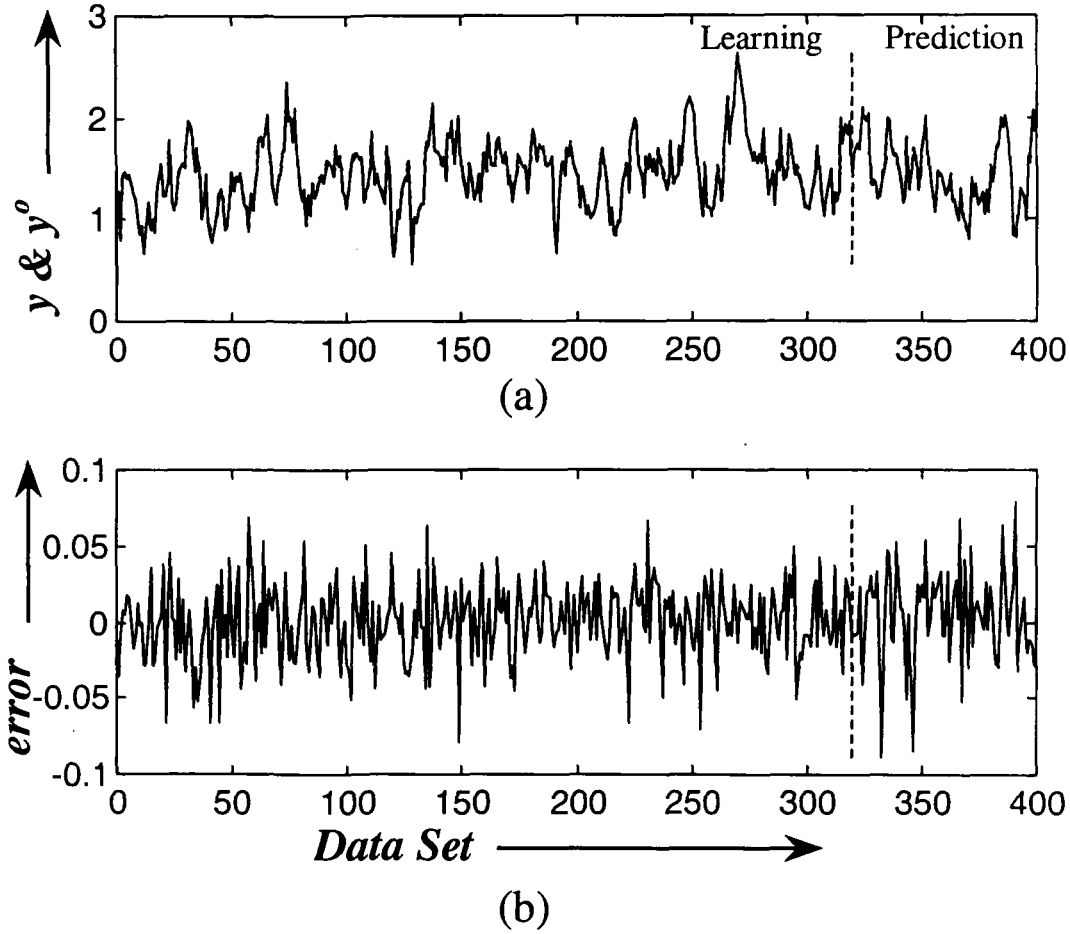


Fig.4.11: Plots of (a) y (————, solid line) and y^o (- - - - - -,dashed line) (b) *model error* for class II GFM of *Example 2.1* noisy data obtain from the initialization of centers based on Modified Mountain clustering method with 3 RBF units.

Example 2.2 Revisited

The GFM of class I and of class II, are tested for their objective function J, validity measure S, SIC and AIC. From Figs. 4.12 and 4.13 we observe that the objective function decreases with modified mountain clustering. In the case of fuzzy curve and fuzzy C-means it reduces and then increases. The performance of the models obtained from all the three methods, with smaller number of rules, are approximately the same. The number of rules appears to be 3 corresponding to the minimum S. The number of rules, for the case of fuzzy curve and fuzzy C-means clustering, is found to be 9 that yields the minimum AIC, and 3 for the case of mountain clustering. Desired output y (solid line) and model output y^o (dashed line) are shown in Fig.4.14(a) and Fig.4.15(a) for the GFM of class I and class II, respectively, obtained from the modified mountain clustering method of initialization with 3 rules. The corresponding *model errors* are plotted in Fig.4.14(b) and Fig.4.15(b). The objective function J of the class I GFM for the training data set is 1.6711×10^{-4} and for prediction is 0.002. The objective function J of the class II GFM for the training data set is 1.5044×10^{-4} and for prediction is 0.0017.

Figure 4.14(b) and Fig.4.15(b) show that the learned model predict very well for the first nine and the last nine percent of the prediction period.

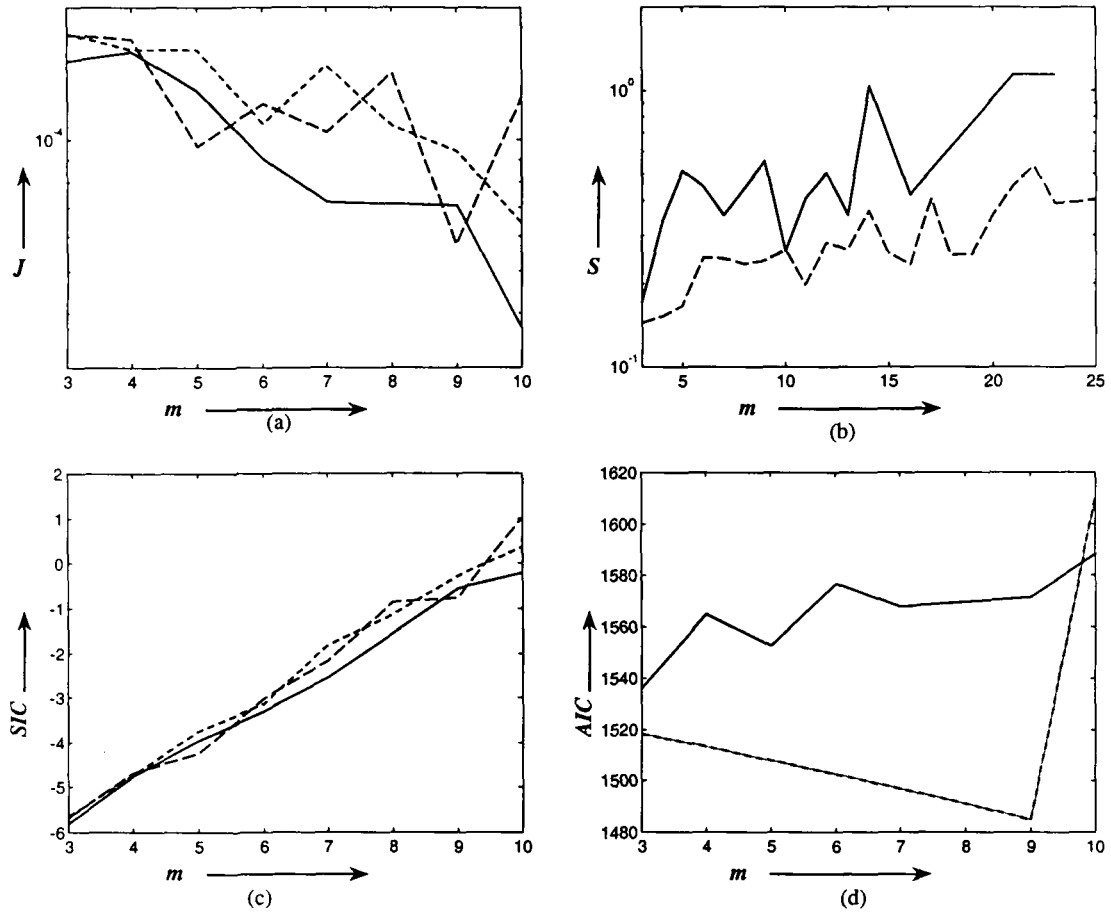


Fig.4.12: Plots of (a) J (b) S (c) SIC (d) AIC for class I GFM of *Example 2.2* w.r.t. different number of RBF units with different methods of initialization of centers, i.e., fuzzy curve (- - - - - - -, short dashed line), Fuzzy C-means clustering (— — — —, long dashed line) and Modified Mountain clustering (————, solid line).

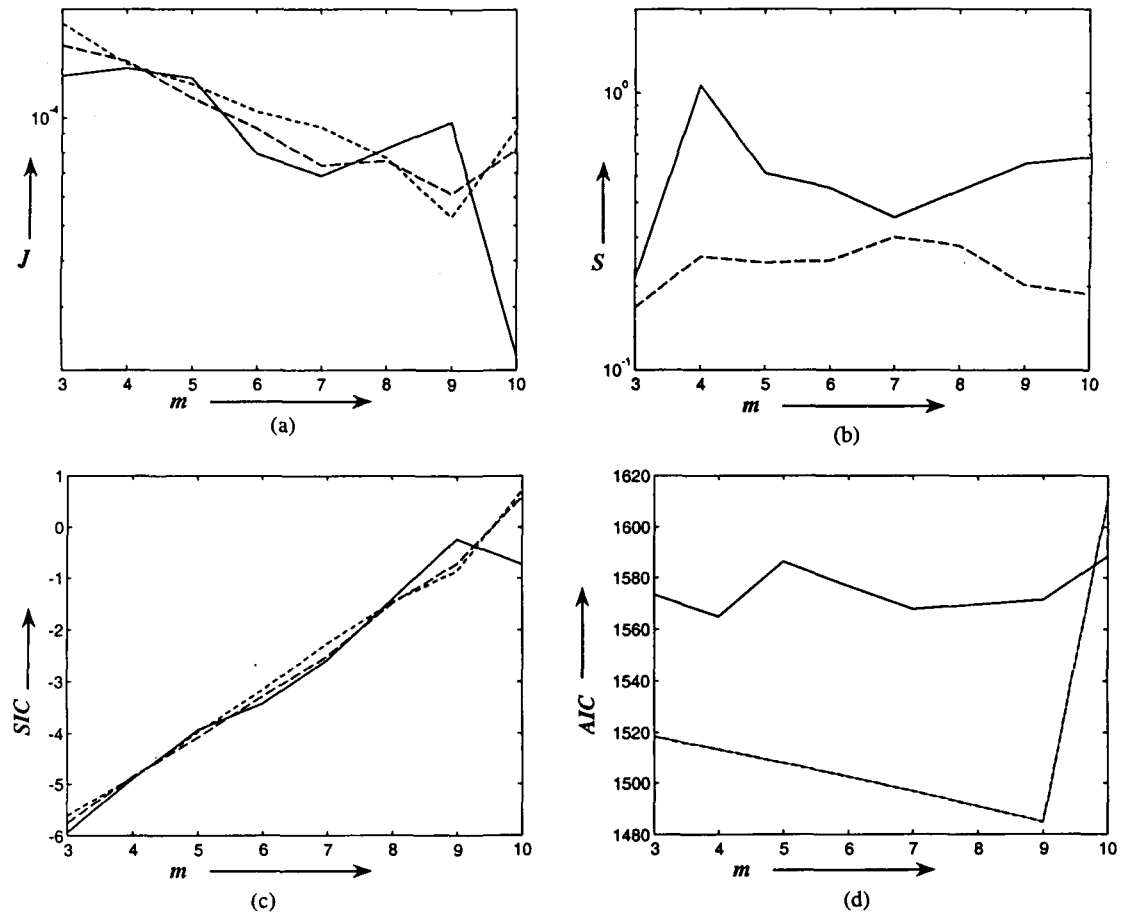


Fig. 4.13: Plots of (a) J (b) S (c) SIC (d) AIC for class II GFM of *Example 2.2* w.r.t. different number of RBF units with different methods of initialization of centers, i.e., fuzzy curve (- - - - - - -, short dashed line), Fuzzy C-means clustering (— — — —, long dashed line) and Modified Mountain clustering (————, solid line).

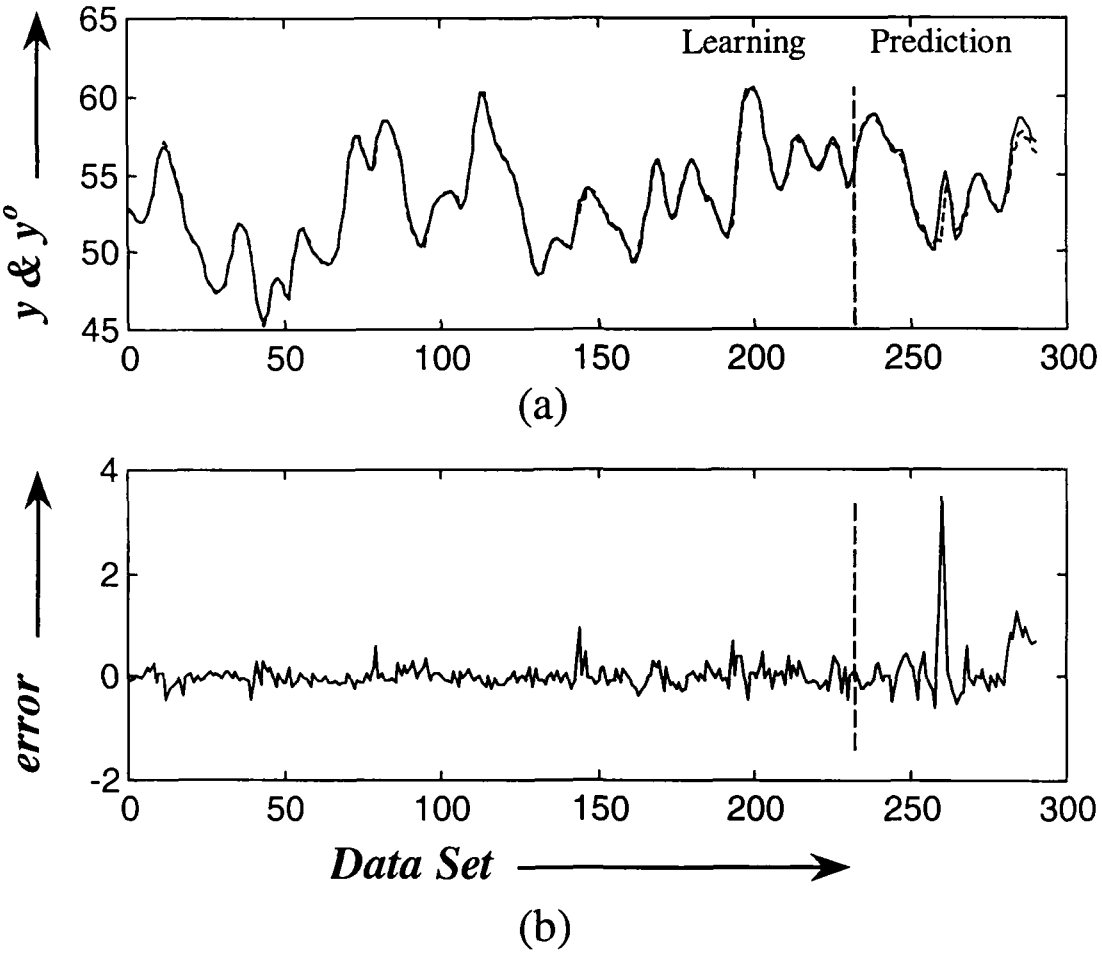


Fig.4.14: Plots of (a) y (————, solid line) and y^o (- - - - - -, dashed line) (b) *model error* for class I GFM of *Example 2.2* obtained from the initialization of centers based on Modified Mountain clustering method with 3 RBF units.

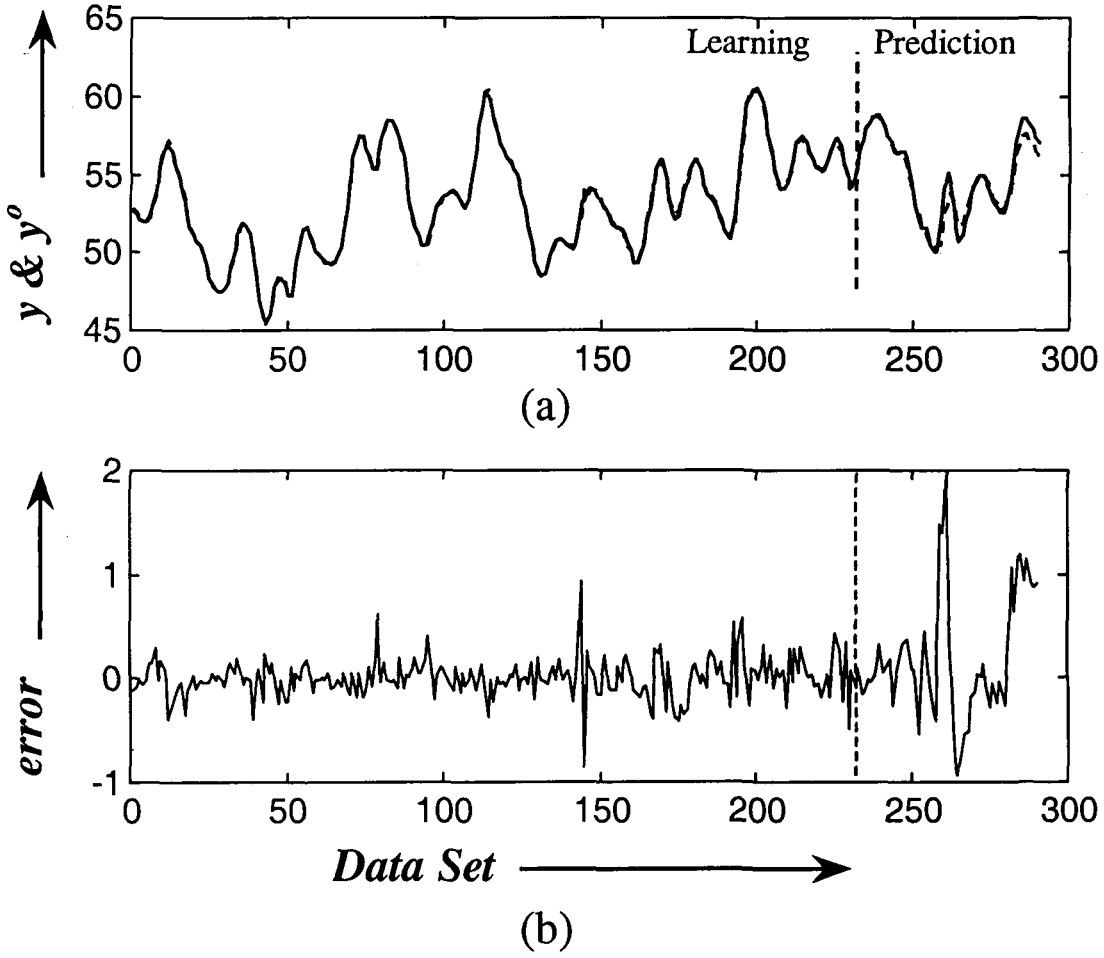


Fig.4.15: Plots of (a) y (———, solid line) and y^o (- - - - - -,dashed line) (b) *model error* for class II GFM of *Example 2.2* obtained from the initialization of centers based on Modified Mountain clustering method with 3 RBF units.

Example 2.3 Revisited

The GFM of class I and of class II, are tested for Human operation at the startup of polymerization plant. Their objective function J, validity measures S, SIC and AIC are plotted in Figs. 4.16 and 4.17. From these figures we observe that the objective function decreases with modified mountain clustering. In the case of fuzzy curve and fuzzy C-means it reduces and then increases. For smaller number of rules performance of the model is approximately the same, for all the three methods. The modified mountain clustering yields a objective function that is almost constant but decreases slightly beyond 3 rules. The number of rules appears to be 5 corresponding to the minimum S since the minimum number of rules obtained from heuristic is 5. The SIC criterion operates well on all the methods yielding 5 rules. The number of rules, for the case of fuzzy curve and fuzzy C-means clustering, is found to be 5 that yields the minimum AIC and 3 for the case of mountain clustering. Desired output y (solid line) and model output y^o (dashed line) are shown in Fig.4.18(a) and Fig.4.19(a) for the GFM of class I and class II, respectively, obtained from the modified mountain clustering method of initialization with 3 rules. The corresponding *model errors* are plotted in Fig.4.18(b) and Fig.4.19(b). The objective function J of the class I GFM is 8.0667×10^{-5}. The objective function J of the class II GFM is 8.0570×10^{-5}.

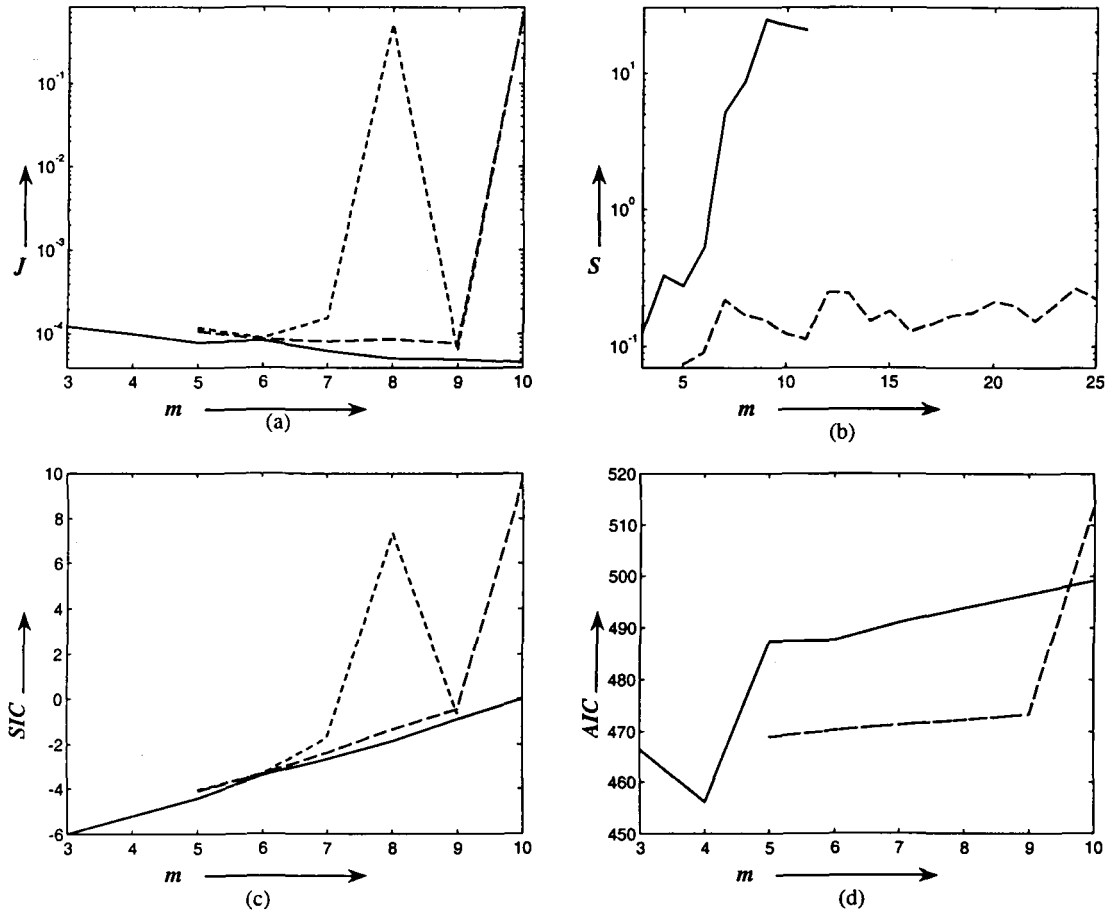


Fig.4.16: Plots of (a) ***J*** (b) *S* (c) ***SIC*** (d) ***AIC*** for class I GFM of *Example 2.3* w.r.t. different number of RBF units with different methods of initialization of centers, i.e., fuzzy curve (- - - - - - -, short dashed line), Fuzzy C-means clustering (— — — —, long dashed line) and Modified Mountain clustering (————, solid line).

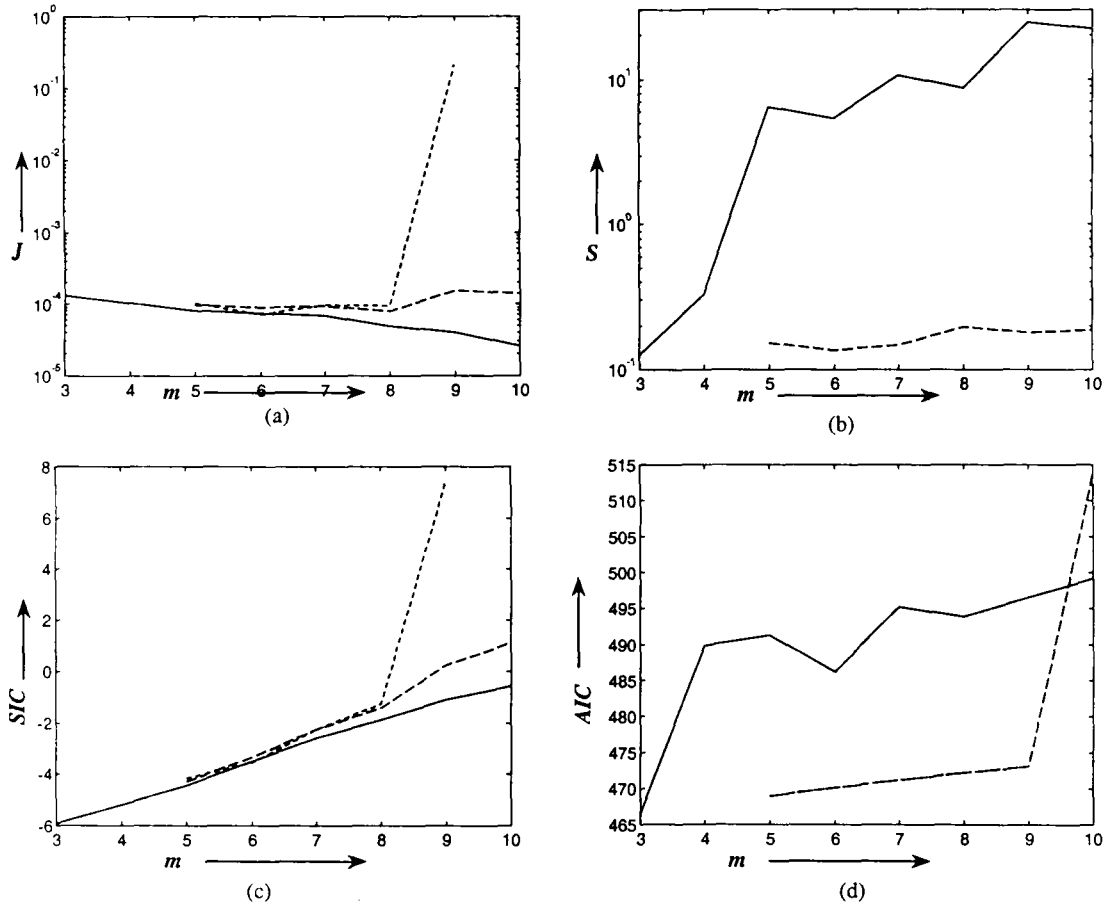


Fig.4.17: Plots of (a) J (b) S (c) SIC (d) AIC for class II GFM of *Example 2.3* w.r.t. different number of RBF units with different methods of initialization of centers, i.e., fuzzy curve (- - - - - - -, short dashed line), Fuzzy C-means clustering (— — — —, long dashed line) and Modified Mountain clustering (————, solid line).

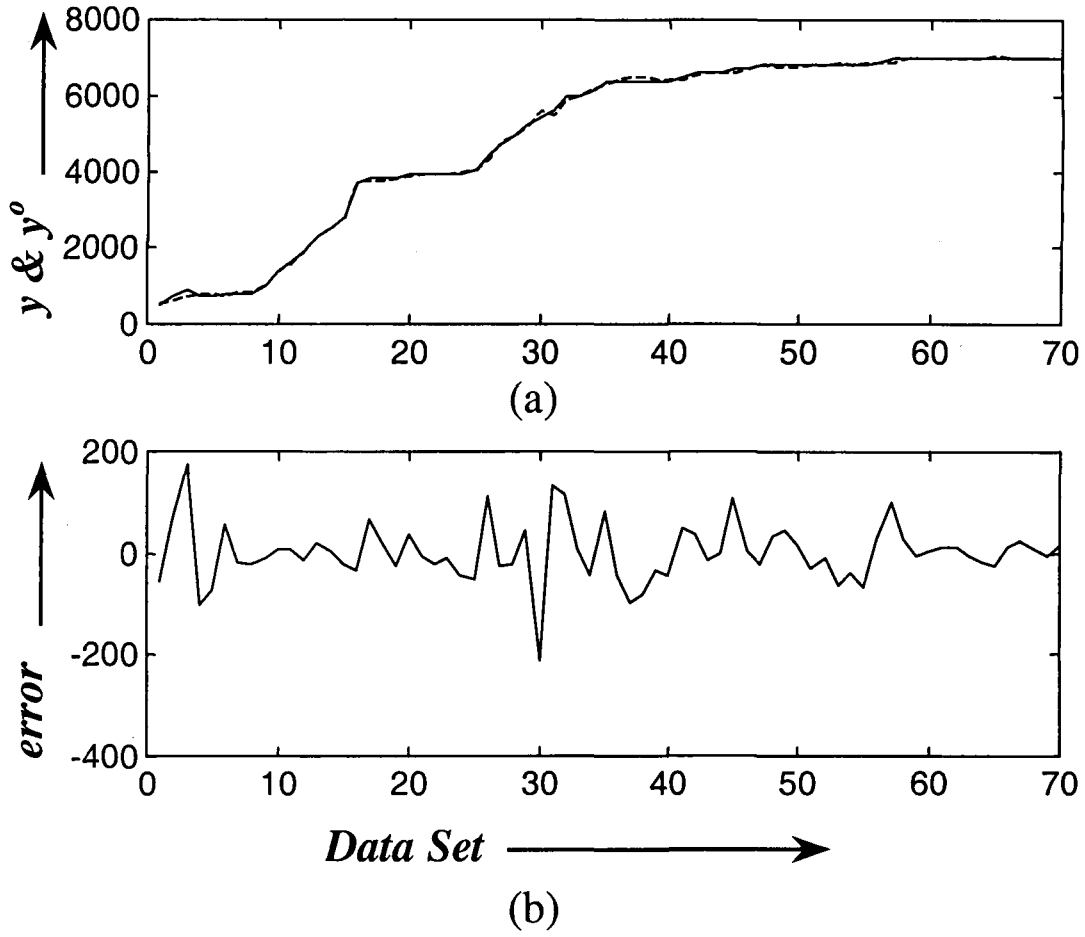


Fig.4.18: Plots of (a) y (————, solid line) and y^o (- - - - - -, dashed line) (b) *model error* for class I GFM of *Example 2.3* obtained from the initialization of centers based on Modified Mountain clustering method with 5 RBF units.

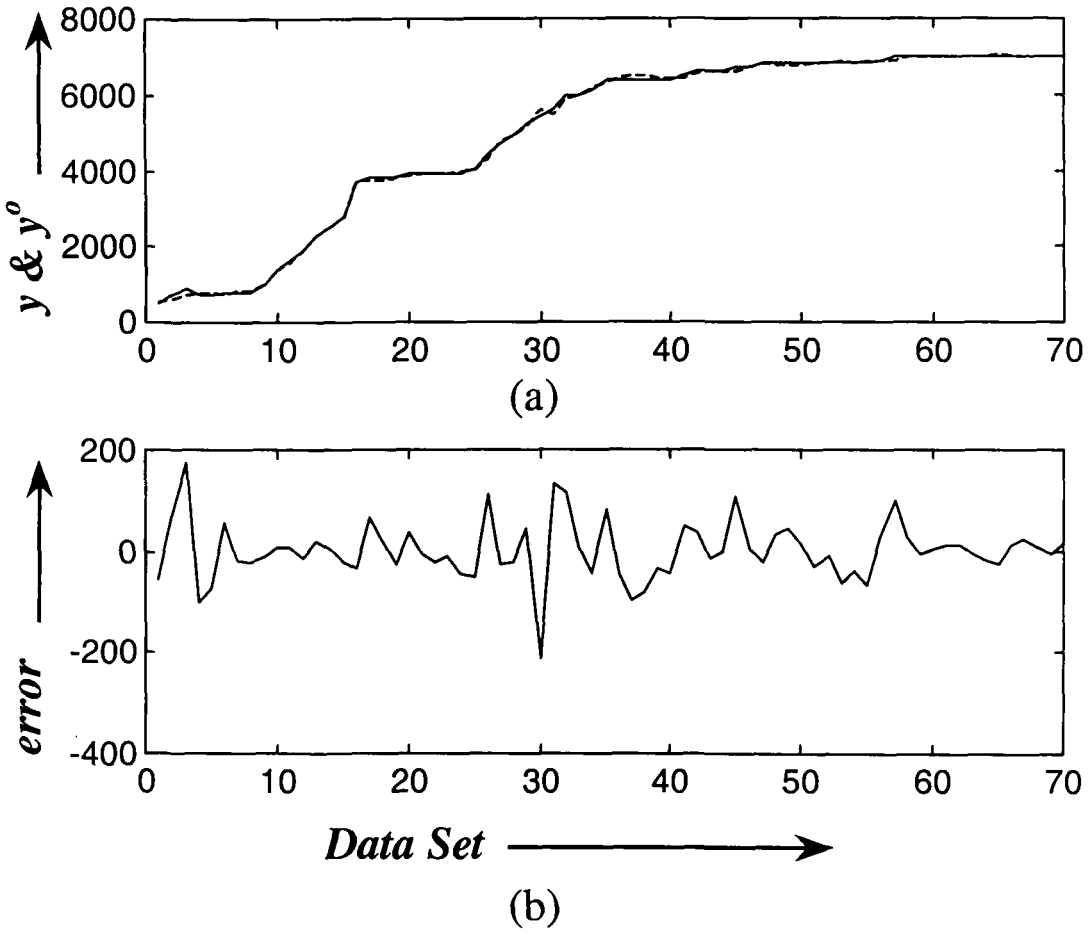


Fig.4.19: Plots of (a) y (————, solid line) and y^o (- - - - - -,dashed line) (b) *model error* for class II GFM of *Example 2.3* obtained from the initialization of centers based on Modified Mountain clustering method with 5 RBF units.

Example 2.4 Revisited

The GFM of class I and of class II, are tested on *Example2.4*. Their objective function J, validity measures S, SIC and AIC are plotted in Figs. 4.20 and 4.21. From Fig.4.20 and Fig.4.21, we observe that the objective function decreases with modified mountain clustering. In the case of fuzzy curve and fuzzy C-means it oscillates within a small range. On the other hand the performance of the model obtained from all the methods is approximately the same, for small number of rules. The number of rules appears to be 4 corresponding to the minimum S for both fuzzy C-means and modified mountain clustering. The SIC criterion operates well on all the methods yielding 4 rules, whereas modified mountain clustering SIC gives the minima at 10 rules for the class I GFM and 9 for the class II GFM. It results due to an over-fitting of the training data. The number of rules, for the case of fuzzy curve and fuzzy C-means clustering, is found to be anywhere between 4 to 9 that yields the minimum AIC, and 4 for the case of mountain clustering, since the minimum number of rules obtained from heuristic is 4. Desired output y (solid line) and model output y^o (dashed line) are shown in Fig.4.22(a) and Fig.4.23(a) for the GFM of class I and class II, respectively, obtained from the modified mountain clustering method of initialization with 3 rules. The corresponding *model errors* are plotted in Fig.4.22(b) and Fig.4.23(b). The objective function J of the class I GFM for the training data set is 0.002 and for prediction is 0.0205. The objective function J of the class II GFMs for the training data set is 0.001 and for prediction is 0.0052.

Figures 4.22(b) and 4.23(b) show that the learned models predict very well for the first nine percent of the prediction period.

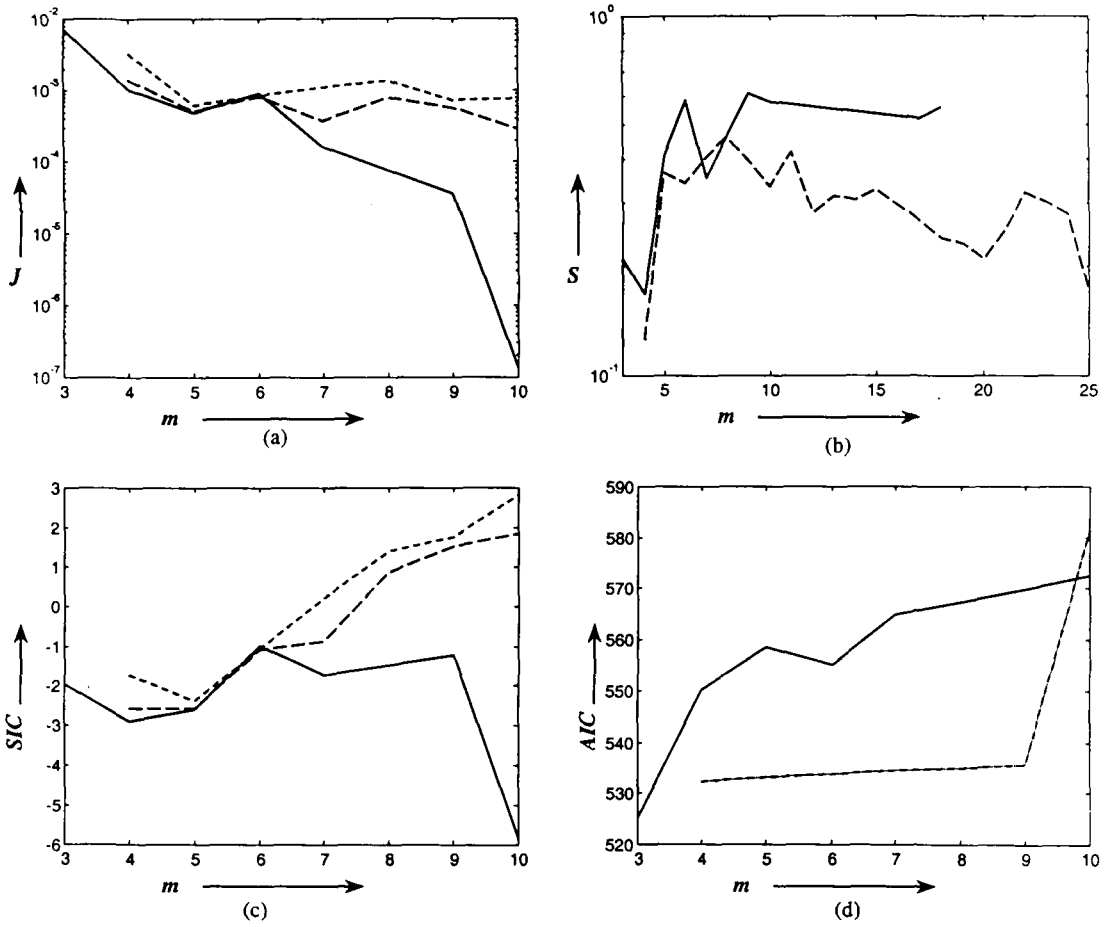


Fig.4.20: Plots of (a) J (b) S (c) SIC (d) AIC for class I GFM of *Example 2.4* w.r.t. different number of RBF units with different methods of initialization of centers, i.e., fuzzy curve (- - - - - - -, short dashed line), Fuzzy C-means clustering (— — — —, long dashed line) and Modified Mountain clustering (————, solid line).

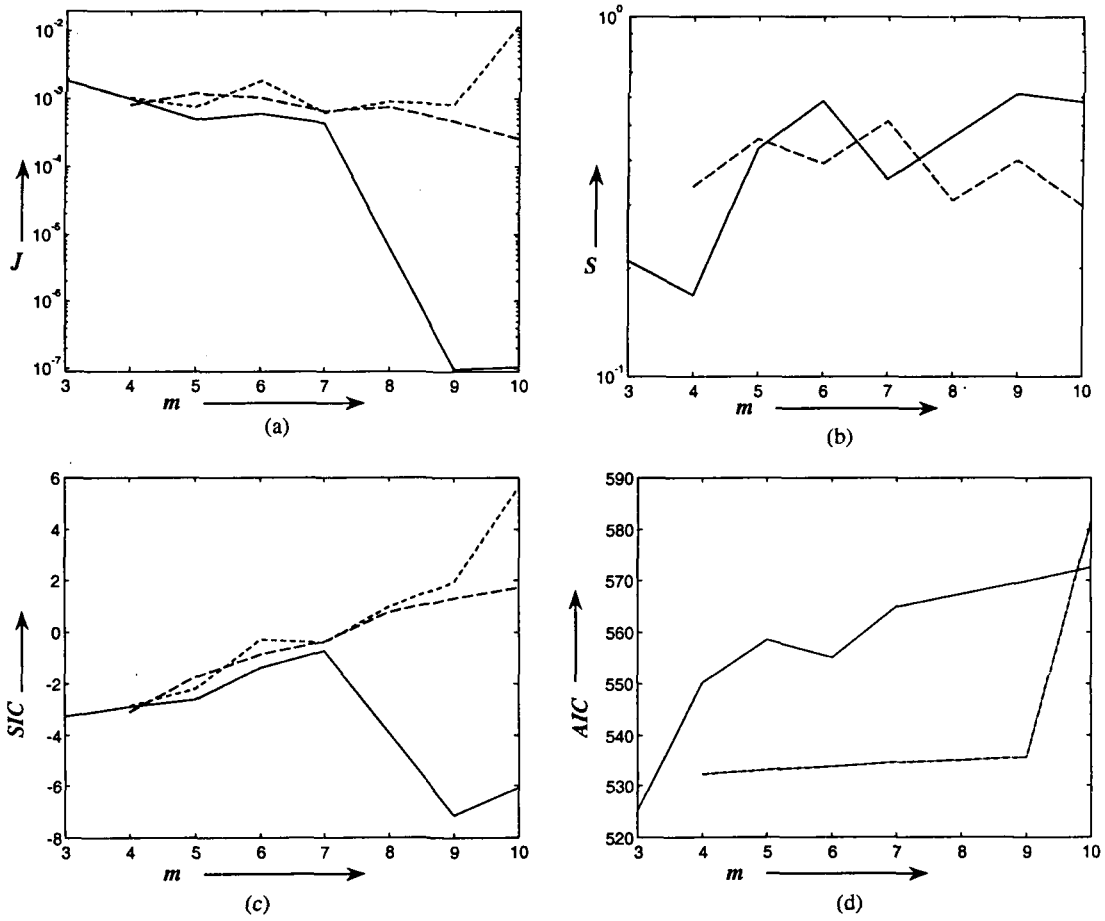


Fig.4.21: Plots of (a) J (b) S (c) SIC (d) AIC for class II GFM of *Example 2.4* w.r.t. different number of RBF units with different methods of initialization of centers, i.e., fuzzy curve (- - - - - - -, short dashed line), Fuzzy C-means clustering (— — — —, long dashed line) and Modified Mountain clustering (————, solid line).

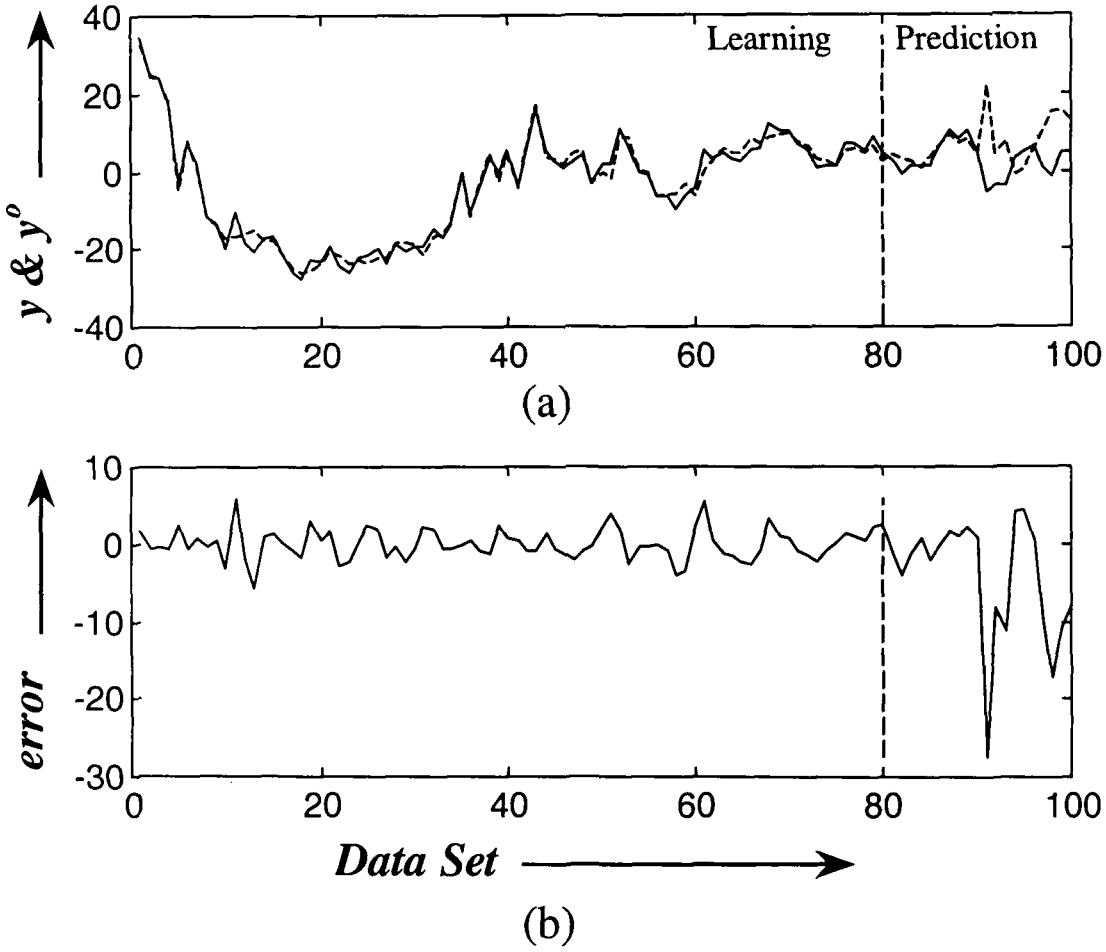


Fig.4.22: Plots of (a) y (————, solid line) and y^o (- - - - - - -, dashed line) (b) *model error* for class I GFM of *Example 2.4* obtained from the initialization of centers based on Modified Mountain clustering method with 4 RBF units.

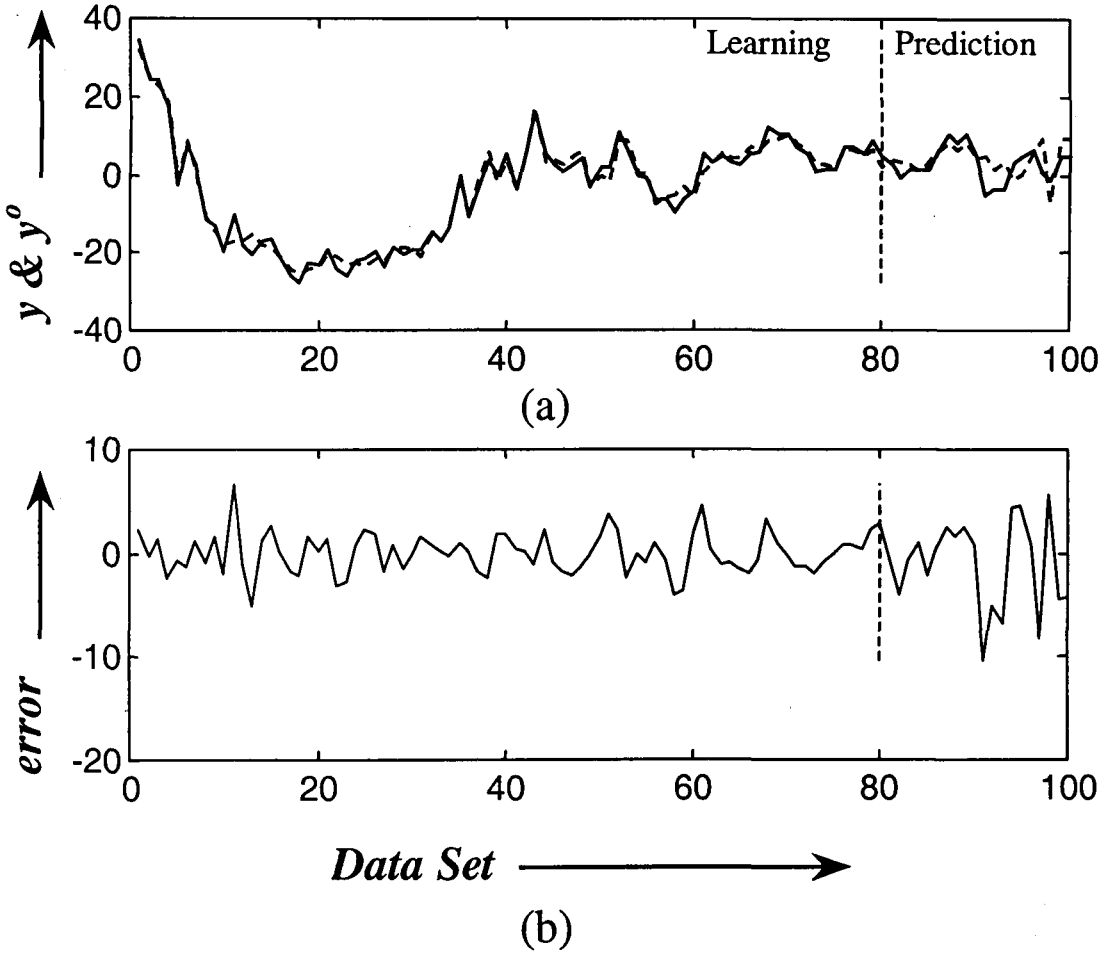


Fig.4.23: Plots of (a) y (————, solid line) and y^o (- - - - - -, dashed line) (b) *model error* for class II GFM of *Example 2.4* obtained from the initialization of centers based on Modified Mountain clustering method with 4 RBF units.

4-6.1 A comparison of Results

TABLE 4-1: List of number of rules with different methods for *Example 2.1* to *Example 2.4*.

Cases			Heuristic based m_{min}	Fuzzy curve			Fuzzy C-means Clustering				Modified Mountain Clustering				Final
				J	*AIC*	*SIC*	*J*	*S*	*AIC*	*SIC*	*J*	*S*	*AIC*	*SIC*	
Example 2.1	Data 1	Class I	3	4	9	3	10	3	9	3	5	3	3	3	3
		Class II	3	4	9	3	9	3	9	3	10	4	10	3	3
	Data 2	Class I	3	9	9	3	10	3	9	3	10	3	3	3	3
		Class II	3	9	9	3	9	3	9	3	10	5	3	3	3
Example 2.2		Class I	3	10	9	3	9	3	9	3	10	3	3	3	3
		Class II	3	9	9	3	9	3	9	3	10	3	4	3	3
Example 2.3		Class I	5	9	5-9	**5**	9	5	5-9	**5**	10	5	5	5	**5**
		Class II	5	8	5-9	**5**	6	6	5-9	**5**	10	6	6	5	**5**
Example 2.4		Class I	4	5	4-9	5	7	4	4-9	5	10	4	4	4	**4**
		Class II	4	5	4-9	4	10	8	4-9	4	9	4	4	4	**4**

The number of rules, corresponding to minimum values of ***J***, *S*, ***AIC*** and ***SIC*** (the plots in Figs 4.4 to 4.23 are all self explanatory)for all the examples with different methods of initialization of GFM, is listed in Table 4-1. The minimum number of rules based on a heuristic derived from fuzzy curve is listed in column 2. The optimum number of rules for each example is listed in the last column. The whole table is divided into two regions shaded by dark gray and light gray colors. The light

gray region shows the same number of rules in each row while the dark gray region has no compatibility with each other. We can make an inference from the light gray region of the table that ***SIC*** gives uniform results for all methods of initialization, ***AIC*** is only suitable for modified mountain clustering, while the validity measure ***S*** gives uniform results. These results, for selection of minimum rule set, match with those obtained using the heuristic based on fuzzy curve, and demonstrate the effectiveness of *SIC* as well as the heuristic. This, in turn demonstrates the utility of modified mountain clustering in giving the optimum number of rules and proper initialization.

4-7 Conclusions

This chapter presents a comparison of three methods, viz., fuzzy curve, fuzzy C-means and modified mountain clustering for the identification of structure of GFM. For ascertaining the extent of trade off between the performance of the fuzzy model and the computation complexity, cluster validity measures, viz., *AIC* and *SIC* have been used. From the simulation results on a few dynamic systems, it has been found that the *SIC* works well on all the three methods yielding the minimum number of rules whereas *AIC* is dependent on the number of data used. The fuzzy curve and fuzzy C-means methods yield more rules with *AIC* than with *SIC*. The modified mountain clustering also yields the minimum number of rules with *AIC*. This study has brought out the distinguishing features of modified mountain clustering over the other two methods and the relevance of *SIC* as a good measure of trade off between the performance and the computational complexity associated with a fuzzy model.

Chapter 5

LEARNING ALGORITHMS FOR GFM

5-1 Introduction

It is part of the folklore of Evolutionary Algorithms (EAs) that hybrids often improve the efficiency of search **[Ibaraki'97]**. Smith **[Smith'85]** and Grefenstette **[Grefenstette'85]** have presented early hybrid algorithms for relatively small prototype systems, and Powell, Tong, and Skolnick **[Powell'89]** are among the first to incorporate hybridization techniques in their commercially viable systems for the design of a gas turbine engine. Davis **[Davis'91]** has perhaps been the foremost advocate of hybridization. It is rare that a serious application is ever undertaken without some kind of global search combined with some specialized search method **[Goldberg'91]**. Along the way, theoreticians have made useful contributions to the state of hybrid knowledge. The distinction made between Baldwinian and Lamarckian learning by Hinton and Nowlan **[Hinton'87]** is particularly important and suggests that a local search method appended to a Global one can have a useful effect without back substituting the genotype corresponding to the termination point of the local search algorithm. A number of practitioners has viewed the probabilistic mixture of different operators as something that should be adapted to achieve various search goals. This theme was picked up in the early

stages of evolution strategies (See **[Back'95]**, for an excellent survey) and the same theme was picked up fairly early in the context of genetic algorithms **[Davis'89; Shaefer'87].** More recently, Lobo and Goldberg **[Lobo'97],** have recast the problem in stochastic automata form, performed empirical tests on a simple test problem, and mapped this theme to the k-armed bandit problem. Whitley **[Whitley'95]** have formulated the precise difference equation and used Markov chain models in the genetic algorithm hybridized with some specified number of steps of local search in either Lamarckian or Baldwinian updates. The formulation is exact; however, the application of results of exact difference or Markov chain equations indicates that, the sledding is tough and usually impedes the kind of systems level understanding we are seeking here. The foregoing contributions are important to the hybridization of learning algorithms.

In this chapter, Section 5-2 presents Least Square Estimate (LSE) framework and Gradient Descent (GD) learning of GFM. The concept of hybridization of Global and Local search is discussed in Section 5-3. Subsequently, implementations of Genetic Algorithm (GA) and Simulated Annealing (SA) are described in Sections 5-4 and 5-5. Simulation results on the examples given in chapter 2 are presented in Section 5-6. Finally conclusions are relegated to Section 5-7.

5-2 Gradient Descent Technique

Fine-tuning of initial rules, obtained from the methods of chapter 4, can be achieved by minimizing the objective function J. It is a function of normalized mean square error with respect to the parameters c_{ki}, a_{ki}, l_{ki}, v_k, and $f^k(x)$. The following form is assumed for J.

$$J = \frac{1}{2.M.y_r} \cdot \sum_{j=1}^{M} e^2(j) \qquad \ldots(5.1)$$

where, $e(j) = y(j) - y^o(j)$ and $y_r = [\mathbf{max}\{y(j)\} - \mathbf{min}\{y(j)\}]^2$.

The reason for using normalized mean square error is that it provides a universal platform for model evaluation irrespective of application and target value specification while selecting an input to the model.

The parameters c_{ki}, a_{ki}, l_{ki}, and v_k are tuned by gradient descent learning. The parameters of local model $f^k(\boldsymbol{x})$ are learned by LSE, while the centroid of consequent membership functions for CRI-model are learned using GD technique. In order to apply LSE, GFM has to be reformulated the in the framework of LSE.

5-2.1 Estimation of GFM by LSE

To evaluate the parameters of local model $f^k(\boldsymbol{x})$, GFM is reformulated in LSE framework as mentioned above, provided that the parameters $\{c_{ki}, a_{ki}, l_{ki} \quad \forall\ i\}$ of membership function of the input hyperspace A^k and index of fuzziness, v_k around the local model, $f^k(\boldsymbol{x})$ are given. The weighted normalized value of input membership function $\mu^k(\boldsymbol{x})$ for j^{th} data is:

$$\hat{\beta}_{jk} = \frac{\hat{\mu}^k(\boldsymbol{x}_j)_{v_k}}{\sum_{l=1}^{m} \hat{\mu}^l(\boldsymbol{x}_j)_{v_l}} \qquad \ldots(5.2)$$

Equation (3.20) can be rewritten in terms of $\hat{\beta}_{jk}$ and local model parameters from eqn.(3.15) as

$$y^o(j) = \sum_{k=1}^{m} \hat{\beta}_{jk} \cdot f^k(x_j) \qquad \forall\, j = 1, \ldots, M$$
$$= \sum_{k=1}^{m} \hat{\beta}_{jk} \cdot \{b_{k0} + b_{k1}\, x_1(j) + \ldots + b_{kn}\, x_n(j)\} \qquad \ldots(5.3)$$

The matrix representation of eqn.(5.3) is

$$\mathbf{Y} = \mathbf{G}\,\mathbf{B} \qquad \ldots(5.4)$$

where,

$$\mathbf{Y}' = [y(1), y(2), \ldots, y(M)]$$

$$\mathbf{B}' = \left[b_{k0}^1, b_{k0}^2, \ldots, b_{k0}^m, b_{k1}^1, b_{k1}^2, \ldots, b_{k1}^m, \ldots, b_{kn}^1, b_{kn}^2, \ldots, b_{kn}^m\right]$$

$$\mathbf{G} = \begin{bmatrix} \hat{\beta}_{11} & \cdots & \hat{\beta}_{1m} & \hat{\beta}_{11}.x_1(1) & \cdots & \hat{\beta}_{1m}.x_1(1) & \cdots & \hat{\beta}_{11}.x_n(1) & \cdots & \hat{\beta}_{1m}.x_n(1) \\ . & \cdots & . & . & \cdots & . & \cdots & . & \cdots & . \\ . & \cdots & . & . & \cdots & . & \cdots & . & \cdots & . \\ . & \cdots & . & . & \cdots & . & \cdots & . & \cdots & . \\ \hat{\beta}_{M1} & \cdots & \hat{\beta}_{Mm} & \hat{\beta}_{M1}.x_1(M) & \cdots & \hat{\beta}_{Mm}.x_1(M) & \cdots & \hat{\beta}_{M1}.x_n(M) & \cdots & \hat{\beta}_{Mm}.x_n(M) \end{bmatrix}$$

Since the training data set, M, is usually greater than the number of parameters, $m(n+1)$, the problem is over-determined and generally there is no exact solution to eqn.(5.4). A least square estimate of $\mathbf{B}$, designed by $\mathbf{B}^*$, which minimizes $\|\mathbf{Y} - \mathbf{GB}\|^2$, is obtained from:

$$\mathbf{B}^* = (\mathbf{G}'\mathbf{G})^{-1}\mathbf{G}'\mathbf{Y} \qquad \ldots(5.5)$$

where, $(\mathbf{G}'\mathbf{G})^{-1}\mathbf{G}'$ is the pseudo-inverse of $\mathbf{G}$ if $\mathbf{G}'\mathbf{G}$ is a non-singular matrix. It is expensive to evaluate this matrix. Moreover, it becomes ill defined when

$\mathbf{G}'\mathbf{G}$ is singular. A sequential formula is used to compute the LSE of $\mathbf{B}$. Let the i^{th} row of $\mathbf{G}$ be g_i and i^{th} element of $\mathbf{Y}$ be y_i, and $\mathbf{B}$ can be calculated iteratively, using the sequential formulas of stable-state Kalman filter **[Takagi'85]**.

$$\left.\begin{aligned} \mathbf{B}_{i+1} &= \mathbf{B}_i + \mathbf{S}_{i+1}\,\mathbf{g}_i'\left(\mathbf{y}_{i+1} - \mathbf{g}_i\,\mathbf{B}_i\right) \\ \mathbf{S}_{i+1} &= \mathbf{S}_i - \frac{\mathbf{S}_i\,\mathbf{g}_i\,\mathbf{g}_i'\,\mathbf{S}_i}{1+\mathbf{g}_i\,\mathbf{S}_i\,\mathbf{g}_i'}, \quad i = 0,1,\ldots,M-1 \\ \mathbf{B}^* &= \mathbf{B}_M \end{aligned}\right\} \qquad \ldots(5.6)$$

The initial values of $\mathbf{B}_o$ and $\mathbf{S}_o$ are set as:

$$\left.\begin{aligned} \mathbf{B}_o &= 0 \\ \mathbf{S}_o &= \xi\,.\,\boldsymbol{I} \end{aligned}\right\} \qquad \ldots(5.7)$$

where, ξ is a large number and $\boldsymbol{I}$ is the identity matrix of dimension $\{m(n+1)\}\times\{m(n+1)\}$.

5-2.2 Gradient descent learning of parameters

In the batch learning scheme employing M-data sets, change in any parameter is governed by the equation:

$$\Delta w(q) = \sum_{j=1}^{M} \Delta_j w(q) + \alpha_m . \Delta w(q-1) - \gamma . w(q) \qquad \ldots(5.8)$$

and the parametric update equation is;

$$w(q+1) = w(q) + \Delta w(q) \qquad \ldots(5.9)$$

where w in eqn.(5.9) may stand for any of the parameters c_{ki}, a_{ki}, l_{ki}, and v_k of all the models and additional parameter b_k of CRI-model. q is q^{th} epoch, α_m is a momentum coefficient in the limits $0 \leq \alpha_m < 1$ (typically $\alpha_m = 0.9$), γ is a decay factor (typically in the range of 10^{-3} to 10^{-6}).

*(a) **Parameter Update Formula***

We apply the gradient descent technique to modify the parameters $b_{ki}, c_{ki}, a_{ki}, l_{ki}$, and v_k. The parameter update formulae for j^{th} data set are as follows:

$$\Delta_j b_k = -\eta \cdot \frac{\partial J}{\partial b_k} \qquad \text{...(5.10)}$$

$$\Delta_j v_k = -\eta \cdot \frac{\partial J}{\partial v_k} \qquad \text{...(5.11)}$$

$$\Delta_j c_{ki} = -\eta \cdot \frac{\partial J}{\partial c_{ki}} \qquad \text{...(5.12)}$$

$$\Delta_j a_{ki} = -\eta \cdot \frac{\partial J}{\partial a_{ki}} \qquad \text{...(5.13)}$$

$$\Delta_j l_{ki} = -\eta \cdot \frac{\partial J}{\partial l_{ki}} \qquad \text{...(5.14)}$$

Applying the chain rule to the above equations, we obtain

$$\Delta_j b_k = -\eta \cdot \frac{\partial J}{\partial e(j)} \cdot \frac{\partial e(j)}{\partial y^o(j)} \cdot \frac{\partial y^o(j)}{\partial b_k}; \qquad \text{...(5.15)}$$

$$\Delta_j v_k = -\eta \cdot \frac{\partial J}{\partial e(j)} \cdot \frac{\partial e(j)}{\partial y^o(j)} \cdot \frac{\partial y^o(j)}{\partial \hat{\beta}_{jk}} \cdot \frac{\partial \hat{\beta}_{jk}}{\partial v_k}; \qquad \text{...(5.16)}$$

$$\Delta_j c_{ki} = -\eta \cdot \frac{\partial J}{\partial e(j)} \cdot \frac{\partial e(j)}{\partial y^o(j)} \cdot \frac{\partial y^o(j)}{\partial \hat{\beta}_{jk}} \cdot \frac{\partial \hat{\beta}_{jk}}{\partial \hat{\mu}^k(x_j)} \cdot \frac{\partial \hat{\mu}^k(x_j)}{\partial \mu^k(x_j)} \cdot \frac{\partial \mu^k(x_j)}{\partial c_{ki}} \quad ...(5.17)$$

$$\Delta_j a_{ki} = -\eta \cdot \frac{\partial J}{\partial e(j)} \cdot \frac{\partial e(j)}{\partial y^o(j)} \cdot \frac{\partial y^o(j)}{\partial \hat{\beta}_{jk}} \cdot \frac{\partial \hat{\beta}_{jk}}{\partial \hat{\mu}^k(x_j)} \cdot \frac{\partial \hat{\mu}^k(x_j)}{\partial \mu^k(x_j)} \cdot \frac{\partial \mu^k(x_j)}{\partial a_{ki}} \quad ...(5.18)$$

$$\Delta_j l_{ki} = -\eta \cdot \frac{\partial J}{\partial e(j)} \cdot \frac{\partial e(j)}{\partial y^o(j)} \cdot \frac{\partial y^o(j)}{\partial \hat{\beta}_{jk}} \cdot \frac{\partial \hat{\beta}_{jk}}{\partial \hat{\mu}^k(x_j)} \cdot \frac{\partial \hat{\mu}^k(x_j)}{\partial \mu^k(x_j)} \cdot \frac{\partial \mu^k(x_j)}{\partial l_{ki}} \quad ...(5.19)$$

where, η is the learning rate > 0. Moreover, the partial derivatives in eqns. (5.15)-(5.19) are as follows:

$$\frac{\partial J}{\partial e(j)} = \frac{e(j)}{M.y_r} \quad ...(5.20)$$

$$\frac{\partial e(j)}{\partial y^o(j)} = -1 \quad ...(5.21)$$

$$\frac{\partial y^o(j)}{\partial \hat{\beta}_{jk}} = f^k(x_j) \quad ...(5.22)$$

$$\frac{\partial y^o(j)}{\partial b_k} = \hat{\beta}_{jk} \quad ...(5.23)$$

$$\frac{\partial \hat{\beta}_{jk}}{\partial v_k} = \frac{\hat{\beta}_{jk}}{v_k}\left(1 - \hat{\beta}_{jk}\right) \quad ...(5.24)$$

$$\frac{\partial \hat{\beta}_{jk}}{\partial \hat{\mu}^k(x_j)} = \frac{\hat{\beta}_{jk}}{\hat{\mu}^k(x_j)}\left(1 - \hat{\beta}_{jk}\right); \quad ...(5.25)$$

The premise parts of all the models along with the consequent part of CRI-model are fine tuned by GD learning while the consequent regression of TS-model and GFM are obtained by LSE technique.

$$\frac{\partial \hat{\mu}^k(x_j)}{\partial \mu^k(x_j)} = 1 \qquad \text{for} \quad \hat{\mu}^k(x_j) = \mu^k(x_j) \qquad \text{...(5.26)}$$

Equation (5.26) holds good for TS-model, class I of both CRI- model and GFM.

$$\frac{\partial \hat{\mu}^k(x_j)}{\partial \mu^k(x_j)} = 2.\{1 - \mu^k(x_j)\} \quad \hat{\mu}^k(x_j) = \mu^k(x_j).\{2 - \mu^k(x_j)\} \quad \text{...(5.27)}$$

Equation (5.27) represents the symmetric triangular membership of the consequent variable, for class II of both CRI-model and GFM

Simplification of eqn.(3.22) leads to:

$$\begin{aligned} \mu^k(x_j) &= \prod_{i=1}^{n} \exp\left(-\left|a_{ki}.\{x_i(j) - c_{ki}\}\right|^{l_{ik}}\right) \\ \text{or} \quad &= \prod_{i=1}^{n} \exp\left(-\left|Z_i^k\{x_i(j)\}\right|^{l_{ki}}\right) \\ \text{or} \quad &= \prod_{i=1}^{n} \exp\left(-\left[O_i^k\{x_i(j)\}\right]^{l_{ki}}\right) \\ \text{or} \quad &= \prod_{i=1}^{n} e^{-S_i^k\{x_i(j)\}} \end{aligned} \qquad \text{...(5.28)}$$

where, $Z_i^k\{x_i(j)\} = a_{ki}.\{x_i(j) - c_{ki}\}$, ...(5.29)

$$O_i^k\{x_i(j)\} = \left|Z_i^k\{x_i(j)\}\right| \qquad \text{...(5.30)}$$

and $$S_i^k\{x_i(j)\} = \left[O_i^k\{x_i(j)\}\right]^{l_{ki}} \qquad \text{...(5.31)}$$

In view of the above, the partial derivatives in eqns.(5.17)-(5.19) are given by:

$$\frac{\partial \mu^k(x_j)}{\partial c_{ki}} = \frac{\partial \mu^k(x_j)}{\partial S_i^k\{x_i(j)\}} \cdot \frac{\partial S_i^k\{x_i(j)\}}{\partial O_i^k\{x_i(j)\}} \cdot \frac{\partial O_i^k\{x_i(j)\}}{\partial Z_i^k\{x_i(j)\}} \cdot \frac{\partial Z_i^k\{x_i(j)\}}{\partial c_{ki}}$$
$$= -\mu^k(x_j) \cdot \left[l_{ki} \left| Z_i^k\{x_i(j)\} \right|^{l_{ki}-1} \right] \cdot \left[sign\left(Z_i^k\{x_i(j)\}\right) \right] \cdot \left[-a_{ki} \right] \quad \ldots(5.32)$$

$$\frac{\partial \mu^k(x_j)}{\partial a_{ki}} = \frac{\partial \mu^k(x_j)}{\partial S_i^k\{x_i(j)\}} \cdot \frac{\partial S_i^k\{x_i(j)\}}{\partial O_i^k\{x_i(j)\}} \cdot \frac{\partial O_i^k\{x_i(j)\}}{\partial Z_i^k\{x_i(j)\}} \cdot \frac{\partial Z_i^k\{x_i(j)\}}{\partial a_{ki}}$$
$$= -\mu^k(x_j) \cdot \left[l_{ki} \left| Z_i^k\{x_i(j)\} \right|^{l_{ki}-1} \right] \cdot \left[sign\left(Z_i^k\{x_i(j)\}\right) \right] \cdot \left[x_i(j) - c_{ki} \right] \quad \ldots(5.33)$$

$$\frac{\partial \mu^k(x_j)}{\partial l_{ki}} = \frac{\partial \mu^k(x_j)}{\partial S_i^k\{x_i(j)\}} \cdot \frac{\partial S_i^k\{x_i(j)\}}{\partial l_{ki}}$$
$$= -\mu^k(x_j) \cdot \left[\left| Z_i^k\{x_i(j)\} \right|^{l_{ki}} \cdot \ln\left(\left| Z_i^k\{x_i(j)\} \right| \right) \right] \quad \ldots(5.34)$$

where,

$$\ln\left(\left| Z_i^k\{x_i(j)\} \right| \right) = \begin{cases} +\text{ve} & \text{for } |x_i(j) - c_{ki}| > \frac{1}{a_{ki}} \left(= \sigma_{ki}\right) \\ \text{zero} & \text{for } |x_i(j) - c_{ki}| = \frac{1}{a_{ki}} \\ -\text{ve} & \text{for } |x_i(j) - c_{ki}| < \frac{1}{a_{ki}} \\ \text{not defined} & \text{for } x_i(j) = c_{ki} \end{cases} \quad \ldots(5.35)$$

We ignore to evaluate the value of $\ln\left(\left| Z_i^k\{x_i(j)\} \right| \right)$ for $x_i(j) = c_{ki}$ since in eqn.(5.34) a term of $\left| Z_i^k\{x_i(j)\} \right|^{l_{ki}}$ becomes zero for $x_i(j) = c_{ki}$. Figure 5.1 depicts the flow chart of gradient descent algorithm, which uses an adaptive learning rate.

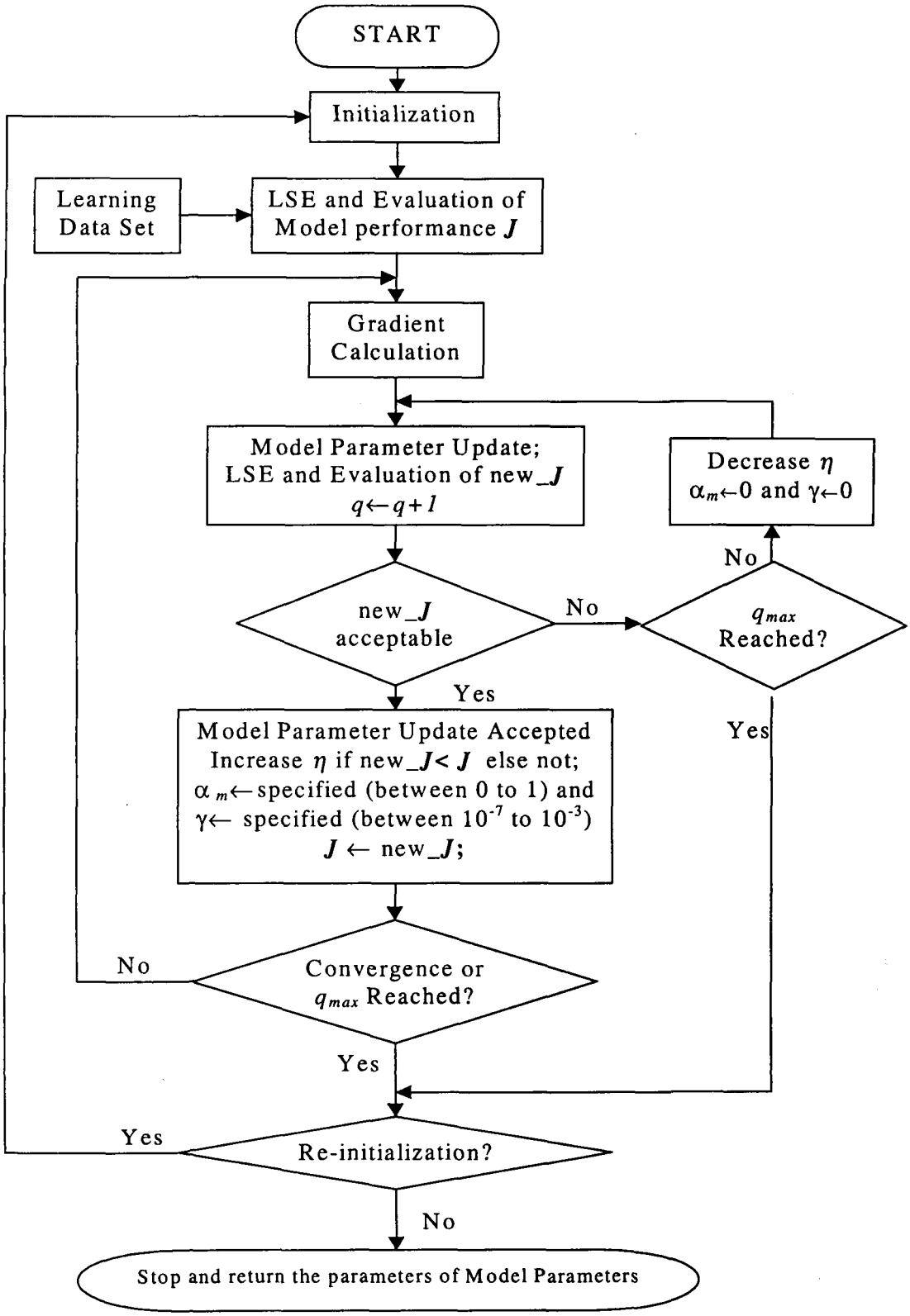


Fig. 5.1: Flow chart of gradient descent algorithm

(b) ***An adaptive learning rate***

A two-phase adaptive scheme, to make the learning rate adaptive, is used in the Gradient Descent Technique. The initial value of learning rate is kept at 0.1 for all applications. In the first phase either it increases or decreases by a factor of "10". When it reaches an appropriate value, in a very few epochs (i.e. < 10), then the second phase starts. This increase or decrease is dependent upon the acceptance or rejection, respectively for updating the parameters. In the second phase, involving the operation $\eta \leftarrow \lambda.\eta$; we choose $\lambda = 1.05$ for the acceptance of parameter updates and $\lambda = 0.7$.for the rejection of the same

(c) ***Convergence test and q_{max}***

When there is no significant change in J over a specified number of epochs ε_q, it is assumed that convergence has reached. The convergence is tested by the formula $\left| J(q-\varepsilon_q) - J(q) \right| < \lambda_J . J(q)$; where $\varepsilon_q = 20$ and $\lambda_J = 0.02$.

So far, we have discussed how the premise and consequent parts of GFM can be obtained using both LSE and GD learning approaches. But if the actual performance of the model is well below the target performance, we go in for hybrid algorithms, which will require an additional learning capability. The entire gamut of hybrid learning is discussed in the following.

5-3 Hybrid Algorithm

This Section considers a framework for creating efficient hybrids of different optimization procedures, viz., genetic algorithm, simulated annealing and gradient descent algorithm to arrive at the global minimal / minimum, while learning GFM.

Though the framework is fairly general, it should work well in other settings for global and local learning.

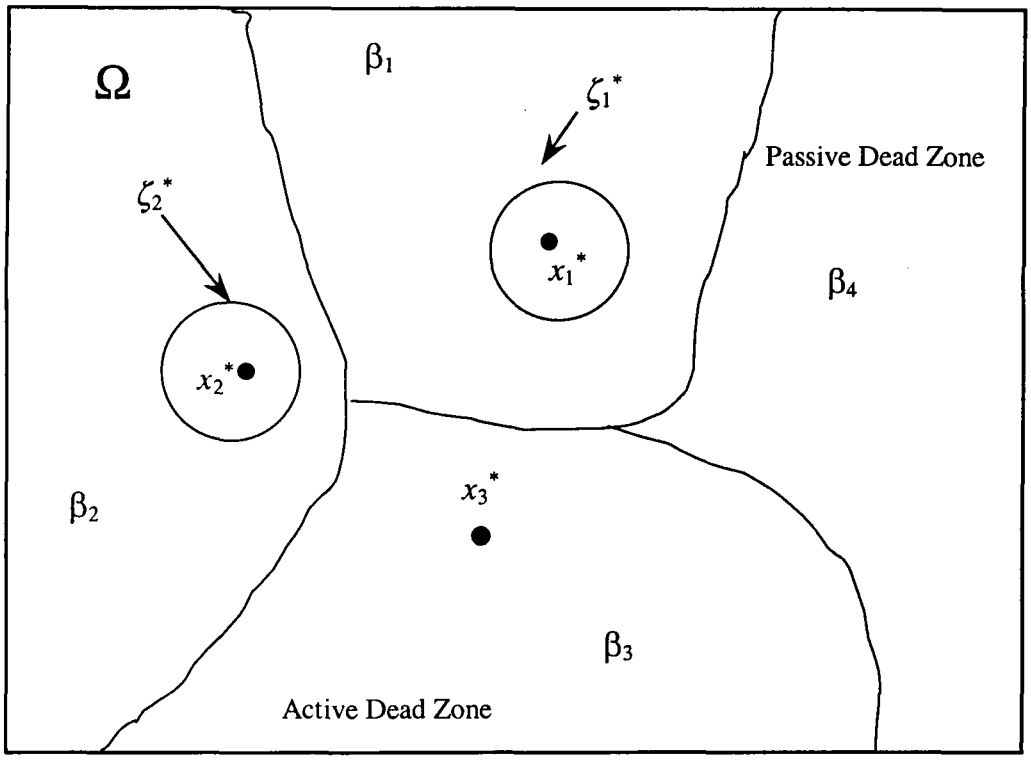


Fig. 5.2: A two-dimensional sketch of a search space depicting target islands ζ_i^*, basins of attraction under local method L to those targets, β_i, and two types of dead zones, active and passive.

Consider a hybrid algorithm (***H***) operating over a solution space Ω of a minimization problem $J : \Omega \rightarrow R$. The hybrid ***H*** coordinates the activities of a global method ***G*** and a local method ***L*** as follows: In each iteration, ***G*** is invoked once (taking unit time, without loss of generality) to generate some new candidate solutions and this is followed by multiple invocations of ***L*** consuming no more than the local allowable time. This process proceeds until either the total allowable time is exceeded or a solution accuracy of target value $J \leq \varepsilon$ is reached.

As regards the solution Ω using $\boldsymbol{H}$, it must be viewed in the context of efficiency. If the solution is to be better than some target value (ostensibly within some $\Delta\varepsilon$ of the global value, $\varepsilon = J^* + \Delta\varepsilon$), we first identify the level set with value ε and subdivide the level set into one or more discrete island targets ζ_i as depicted in Fig.5.1. Then we explore how $\boldsymbol{G}$ and $\boldsymbol{L}$ can be combined to lead us to one of the targets. Basically, we search the union of discrete island targets ζ_i (the solution space) by using local method $\boldsymbol{L}$. This is explained in the following.

The first thing to understand is that $\boldsymbol{G}$ may be successful on its own. Let us call the union of the targets as the global region, $\boldsymbol{R}_G = \bigcup_i \zeta_i$. Outside $\boldsymbol{R}_G$, $\boldsymbol{G}$ requires $\boldsymbol{L}$, but to describe this interaction with the $\boldsymbol{L}$, we need the ideal search space. Recognizing that local search amounts to a dynamic system, usually the solution process works by iterating from some starting position to some solution. Usually $\boldsymbol{L}$ converges toward a fixed-point solution x_i^* in the basins of attraction β_i as shown in Fig. 5.2. We should note that targets might contain more than a single solution (as proved in Section 3-4.4). Accordingly, it makes sense to partition the targets and basins of attractions along the solution lines.

In difficult problems, hitting the targets with $\boldsymbol{G}$ directly is unlikely, but the chance of hitting one of the basins and converging to a target with $\boldsymbol{L}$ is quite good. Suppose instead of landing in the global zone or in any one of the tractable basins, $\boldsymbol{G}$ makes the search at points in the space that, do not lead to the global zone under local search. This necessitates the consideration of two cases. $\boldsymbol{G}$ may land in a basin such as β_3 in which local search leads to a solution that does not meet the criterion ($\boldsymbol{J} \leq \varepsilon$). Worsely, $\boldsymbol{G}$ may move to a basin (β_4) where $\boldsymbol{L}$ fails to converge in the maximum allowable time units. We call both these regions as dead zones. A type I dead zone is distinguished in $\boldsymbol{H}$ as the one

that converges to a solution, but the solution is inadequate. A type II dead zone does not lead to convergence of solution and if the solution were permitted to continue, it would consume the specified computational time. In other words, the dead zone is what is left over when the global zone and tractable basins are removed. Simply saying the dead zones are the places where we do not wish G to land.

We test the hybrid of (i) GA with gradient descent, and (ii) simulated annealing with gradient descent. GA and SA perform the global search in the space of reduced dimension while gradient descent is strictly a local search. Based on the Theorem 3.1 and the results of the associated example, we perform the global search in the space of parameter matrix $c \in \Re^{n \times m}$ and the local search in the space of parameter $S \in \Re^{(4n+2) \times m}$.

As mentioned in the hybridization, our aim is to make local search by means of additional learning. We now discuss some local learning methods using genetic algorithm and simulated annealing.

5-4 Genetic Algorithm and Gradient Descent

Before going to discuss the hybridization scheme, we briefly review the functioning of GA.

5-4.1 A Brief Review of Genetic Algorithm

The basic theory of GA can be found in **[Goldberg'89]**, and in this Section we briefly discuss what are the components of GA and how do they function in the solution process.

Suppose we are seeking to find a solution to some problem. To apply a genetic algorithm to that problem, the first thing to do is to encode the problem into artificial chromosomes. These artificial chromosomes can be the strings of 1's and 0's, or the parameter lists, or even the complex computer codes, but the key thing to keep in mind is that the genetic machinery will manipulate a finite representation of the solutions, not the solutions themselves. The second thing to do to solve a problem is to have some means of discriminating good solutions from bad ones. This can be as simple as having a human intuitively choose better solutions, or it can be an elaborate computer simulation or a model that helps determine the quality of a solution. But the idea is to ascertain a solution's relative *fitness* to purpose by some means. The genetic algorithm will use these very means to guide the evolution of future generations.

Having encoded the problem in terms of chromosomes and having devised a means of discriminating good solutions from bad ones, we prepare to evolve solutions to our problem by creating an initial *population* of encoded solutions. The population can be created randomly or by using prior knowledge of possible good solutions, but either way GA will search from a population, not from a single point.

There are various types of operators that are used in GAs, but quite often (1) *selection,* (2) *recombination* and (3) *mutation.* The selection and genetic operators can process the population iteratively to create a sequence of populations that will hopefully contain more and more good solutions to our problem over a time.

Simply stated, selection operator allocates greater survival to better individuals. This is what is known as the survival of the fittest mechanism, which we impose on our solutions. This can be accomplished in a variety of ways. Weighted *roulette wheels* can

be spun, local *tournaments* can be held, various ranking schemes can be invoked, but whatever we do, the main task is to seek better solutions over worse ones. Of course, if we were to only choose better solutions repeatedly from the original database of initial solutions, we would expect the population to contain the best solution of the first generation. However, simply selecting the best is not enough, and some means of creating new, possibly better individuals must be found. This is where the mechanisms like recombination and mutation come into play.

Recombination is a genetic operator that combines bits and pieces of parental solutions to form new, possibly better offspring. Again, there are many ways of accomplishing this. Achieving desirable performance does depend on getting the recombination mechanism designed properly; but the primary concern is to see that the offspring under recombination will not be identical to any particular parent, so we combine the parental traits in a novel manner. Recombination by itself is not all that useful, because a population of individuals processed under repeated recombination alone will undergo what amounts to a random shuffling of existing traits.

As against recombination, which creates a new individual by combining the traits of two or more parents, mutation acts by simply modifying a single individual as shown in Fig. 5.3. There are many variations of mutation, but the main constraint is that the offspring must have traits identical to the individual parental traits except that the operator may make one or more changes to an individual's traits. By itself mutation represents a ``random walk" in the neighborhood of a particular solution. If applied repeatedly over a population of individuals, we might expect the resulting population to be indistinguishable from the one created at random.

5-4.2 Components of GA

There are various possible combinations of different components to construct a GA. A detailed description of different combinations can be found in **[Goldberg'89, Davis'91].** Here we shall give a brief description of the combinations used in this thesis.

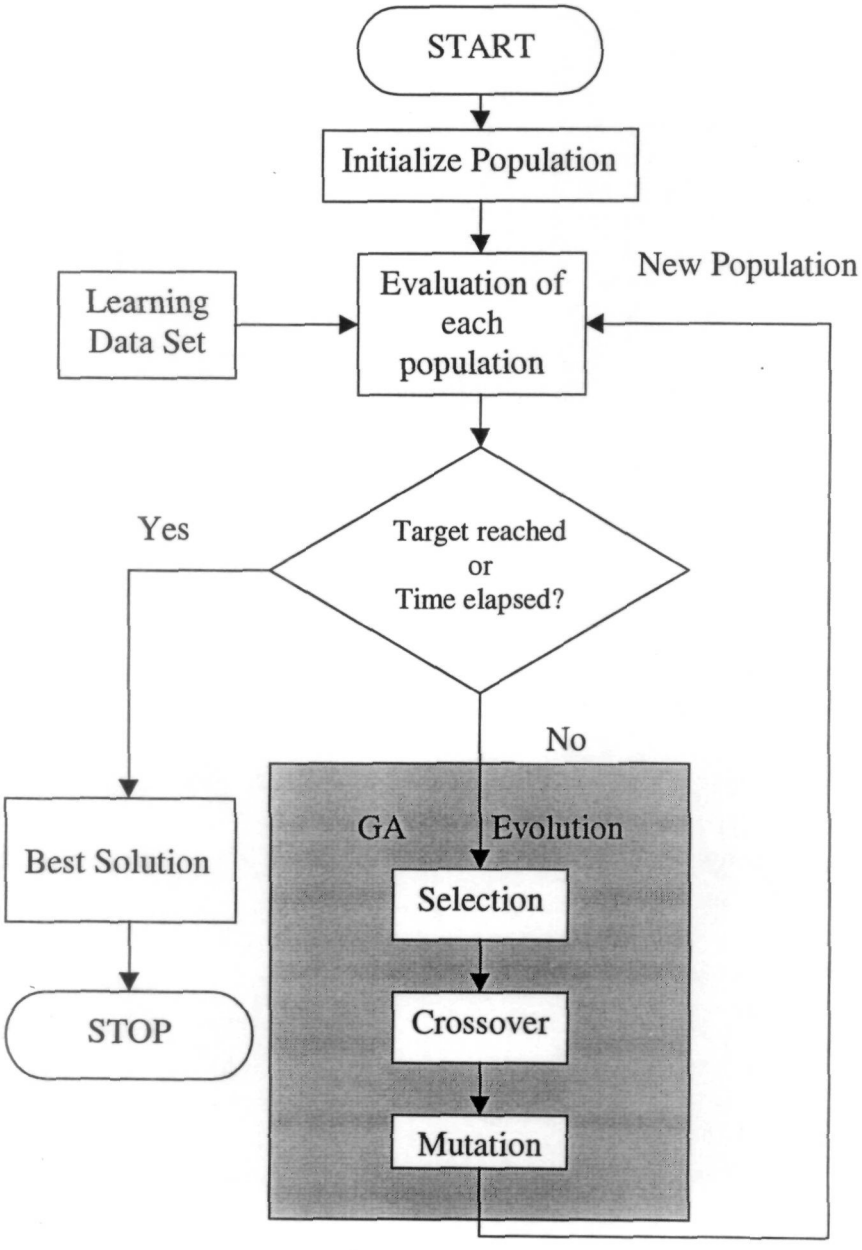


Fig. 5.3: Flow chart of a standard Genetic Algorithm.

(a) ***Solution Representation (Encoding & Decoding)***

Binary string representation scheme is used to perform GA evolution. Since we are required to encode the center matrix $\boldsymbol{c} \in \Re^{n \times m}$., we normalize each element of $\boldsymbol{c}$ in the search space by the span, i.e., $(c_{ki,\max} - c_{ki,\min})$, to yield $c_{ki,\text{nor}} = (c_{ki} - c_{ki,\min})/(c_{ki,\max} - c_{ki,\min})$. The decimal value, $decimal(c_{ki,2})$, of each element of $\boldsymbol{c}$, for the binary string of length h, is obtained from the relation: $decimal(c_{ki,2}) = c_{ki,\text{nor}}/(2^h - 1)$ and the resolution of the binary string is $(c_{ki,\max} - c_{ki,\min})/2^h$. Now the $decimal(c_{ki,2})$ is converted into the binary string by adding sufficient number of "0s" on the left side of the string in order to complete the specified string length, i.e. h. With this, the total binary length for each solution (chromosome) is $h \times m \times n$. A fixed binary length h is taken as 10 for each c_{ki}. Further, it may be noted that gradient descent learning takes care of resolution interval, inherited by the genetic coding, if the solution lands in the region of basin.

The solution must be decoded before it is evaluated. Steps involved in decoding are (i) separate the string of length h corresponding to each c_{ki}, (ii) convert this string into decimal value $decimal(c_{ki,2})$ and (iii) obtain the value of c_{ki} from $decimal(c_{ki,2})$ by the following formula.

$$c_{ki} = c_{ki,\min} + decimal(c_{ki,2}) \frac{(c_{ki,\max} - c_{ki,\min})}{(2^h - 1)} \qquad \ldots(5.36)$$

(b) Initialization

A specified number of solution strings of 0's and 1's is generated randomly as an initial population. We can also feed some of the solutions of chapter 4 as the initial population.

(c) Evaluation function

Each decoded solution represents a model. An evaluation (fitness) function is defined to evaluate the degree of fitness of all the models with respect to learning data set. Our goal is to minimize the objective function J defined by eqn.(5.1). Since, GA is used strictly for the maximization problems, without loss of generality, fitness function is defined as the reciprocal of the objective function J. So, minimization of J and maximization of fitness function are equivalent.

(d) Selection, Crossover, Mutation and Reproduction

Weighted *roulette wheels* approach is opted for parent selection. In the parent selection care, has to be taken such that two identical parents should not crossover, to prevent the production of two identical children similar to their parent, whose fitness has already been tested.

The two-point crossover is used for the reproduction of offspring. The two points for the crossover in the chromosome string are selected randomly. The probability of crossover is set at $p_c = 0.8$.

The number of mutations in a solution is randomly selected with a very small value of probability, i.e., $p_m = 0.02$.

The technique of generational replacement without duplication is used for reproduction, and to test for a new solution. In this technique, all the solutions of one generation are replaced by the solutions of the next generation.

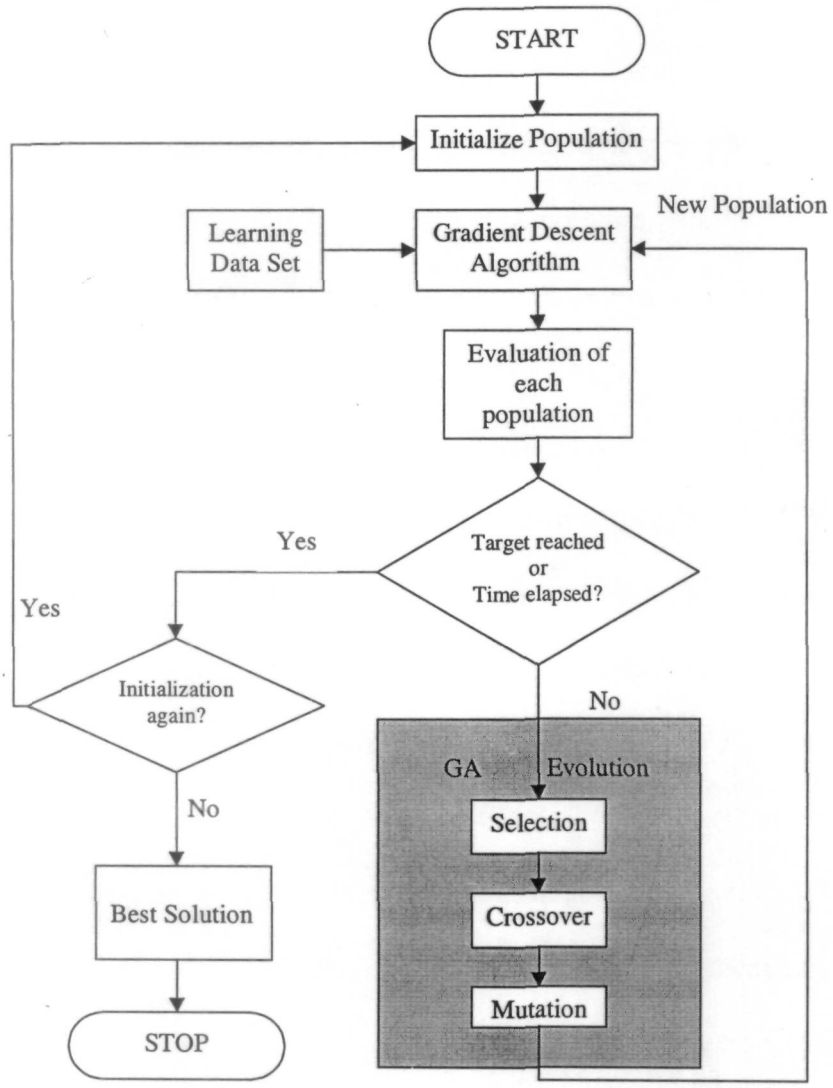


Fig. 5.4: Flow chart of a hybrid of Genetic Algorithm and Gradient Descent.

(e) ***Hybridization Scheme***

Hybridization of GA with Gradient descent is depicted in Fig.5.4. Since GA works with multiple solutions, we apply the Gradient descent to all the solutions in each generation, and keep track of the best solution.

5-5 Simulated Annealing and Gradient Descent

Simulated annealing was first applied to an optimization problem in **[Kirkpatrick'83]** for modular placement. It is a random walk method on the error surface. It is very time consuming but yields excellent results. It also provides an excellent heuristic for solving any combinatorial optimization problem, such as the traveling salesman problem **[Randelman'86]**.

5-5.1 A Brief Review of Simulated Annealing

In simulated annealing all the moves that result in reduction in the objective function J are accepted. Moves that result in an increase in J are accepted with a probability that decreases with the very increase in J. A parameter T, called the *temperature,* is used to control the probability of accepting the increased J. Higher values of T cause more such moves to be accepted. In most implementations of this algorithm, the acceptance probability is given by $\mathbf{exp}(-\Delta J/T)$, where $\Delta J = \mathbf{new}\_J - J$, i.e., increases in the objective function. In the beginning the temperature is set to be very high so most of the moves are accepted. Then the temperature is gradually decreased so the moves with increased J have less chance of acceptance. Ultimately, the temperature is reduced to a very low value, so that only moves with decreased J are accepted and the algorithm converges to a low cost configuration.

A typical simulated annealing hybrid algorithm with gradient descent is given as under:

```
Algorithm

PROCEDURE
        Initialize;
        Start with an Initial solution;
        WHILE1 stopping_criterion ( outerloop_count, temperature, J ) =FALSE
                WHILE2 inner_loop_criterion=FALSE
                        new_solution ← grad_descent{ perturb (solution) };
                        ΔJ ← evaluate(new_solution, solution);
                        IF1 ΔJ < 0 THEN solution ← new_solution [move accepted]
                        ELSE IF2  accept(ΔJ ,temperature) > random(0,1)
                                        THEN solution ← new_solution [move accepted]
                                ENDIF2
                        ENDIF1
                END WHILE2
                temperature ← schedule(outerloop_count, temperature);
                outerloop_count ← outerloop_count+1;
        END WHILE1
END
```

5-4.2 Components of SA

The above hybrid algorithm includes some functions which are described as under:

Perturb (solution) generates a random variation to the current solution. The number of elements and which element of S matrix to be perturbed is decided randomly. The amount by which the element is to be perturbed is selected randomly. Perturbation to each element is accepted when the perturbed value of the element remains within the pre-specified limit, otherwise rejected.

Stopping_criterion (outerloop_count, temperature, J) terminates the algorithm when the any of the parameters, i.e., temperature, the outerloop_count or J reach a threshold value. The $WHILE_1$ loop stops if any one of the following conditions is met.

- outerloop_count > max_outerloop, and max_outerloop is set to 120.
- temperature < 1×10^{-8} (How to choose the temperature limit is discussed in ***Schedule***)
- $J \leq \varepsilon$

inner_loop_criterion It decides the number of trials at each temperature. The $WHILE_2$ loop stops if any of the following conditions is met.

- $J \leq \varepsilon$
- The innerloop_count becomes greater than the assigned max_inloop. The innerloop_count represents the number of perturbations at a particular temperature (temperature is a function of outer loop_count). Since at high temperatures, the probability of accepting a perturbed solution with higher value of J is high, so the number of perturbations should be less. Conversely, at low temperature very few moves with higher J will be accepted so the number of perturbations at low temperature should be more. The value of max_inloop varies from a minimum of 2 to a maximum of 200. Figure 5.5 shows the plot of max_inloop vs. outerloop_count.

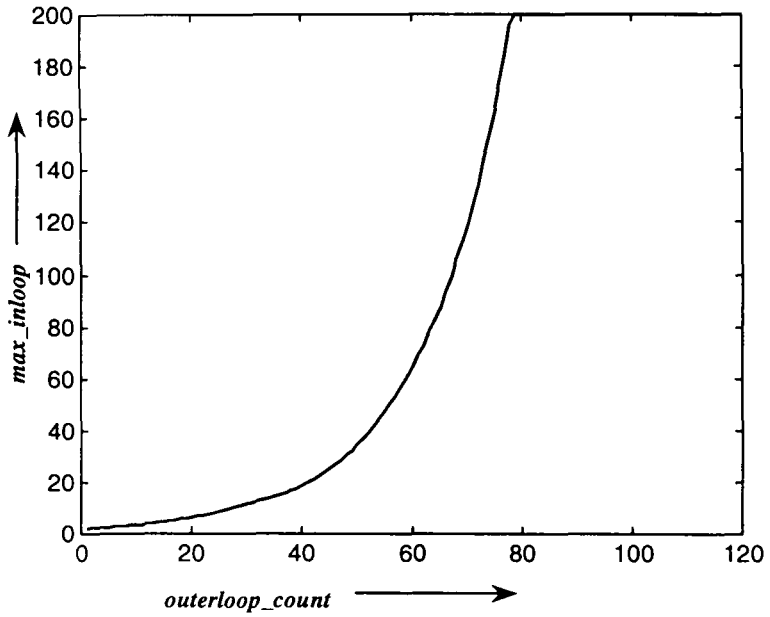


Fig. 5.5: Plot of max_inloop vs outerloop_count.

Evaluate (new_solution, solution) evaluates the change in objective function as $\Delta J = \text{new}\_J - J$.

Accept $(\Delta J, T)$ is the probabilistic acceptance function that is called when ΔJ is positive. It determines whether to accept a move or not, depending upon ΔJ and temperature T. We have used an exponential function, $\exp(-\Delta J/T)$ for the probability.

Schedule (outerloop_count, temperature) is the temperature schedule, which gives the next temperature as a function of outerloop_count and the previous temperature. Figure 5.6 shows the annealing schedule (temperature vs. outerloop_count).

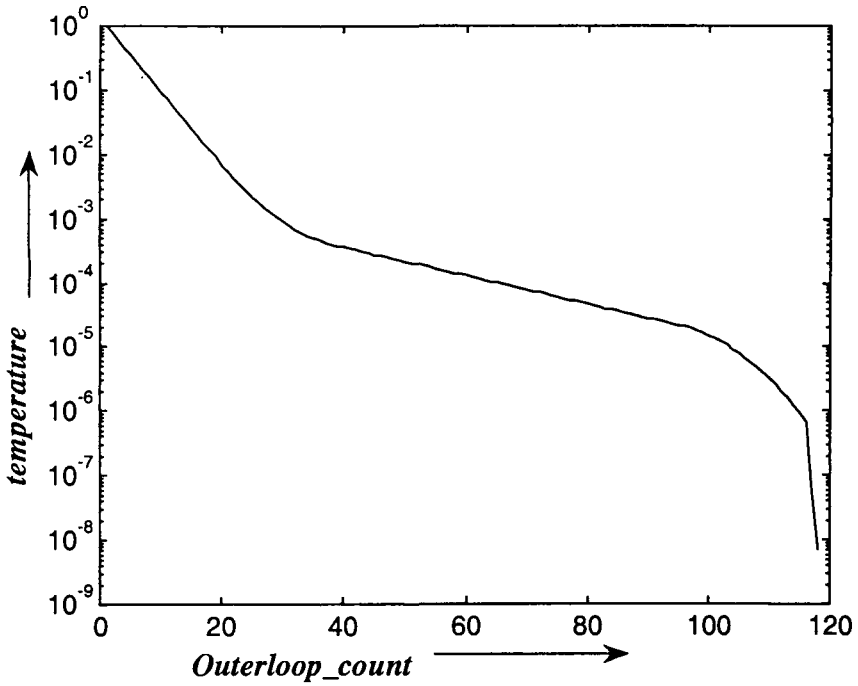


Fig. 5.6: Annealing Schedule as a function of outerloop_count.

There is no fixed rule about the initial temperature, generally it is kept at very high value. In order to choose a value for the initial temperature, we need to discuss on the range of J and the probabilistic function $\mathbf{P}_{accept} = \exp(-\Delta J/T)$. Since the objective function J is the per unit mean square error and its range is $0 \leq J \leq 1$, so the range of ΔJ

is $0 \leq \Delta J \leq 1$. Hence a good choice for initial temperature is "1" with an acceptance probability of 0.3679 for $\Delta J = 1$, and the final temperature is 10^{-8} corresponding to the target value of objective function, i.e., $\varepsilon = 2\times10^{-4}$. Plots of $\mathbf{P}_{accept}$ vs. outerloop_count and $\mathbf{P}_{accept}$ vs. temperature corresponding to different values of ΔJ are shown in Fig.5.7 and Fig.5.8 respectively. Figure 5.9 shows a band of ΔJ for $\mathbf{P}_{accept}$ in the range $0.001 \leq \mathbf{P}_{accept} \leq 0.99$ corresponding to the temperature range, $1 \geq T \geq 10^{-8}$.

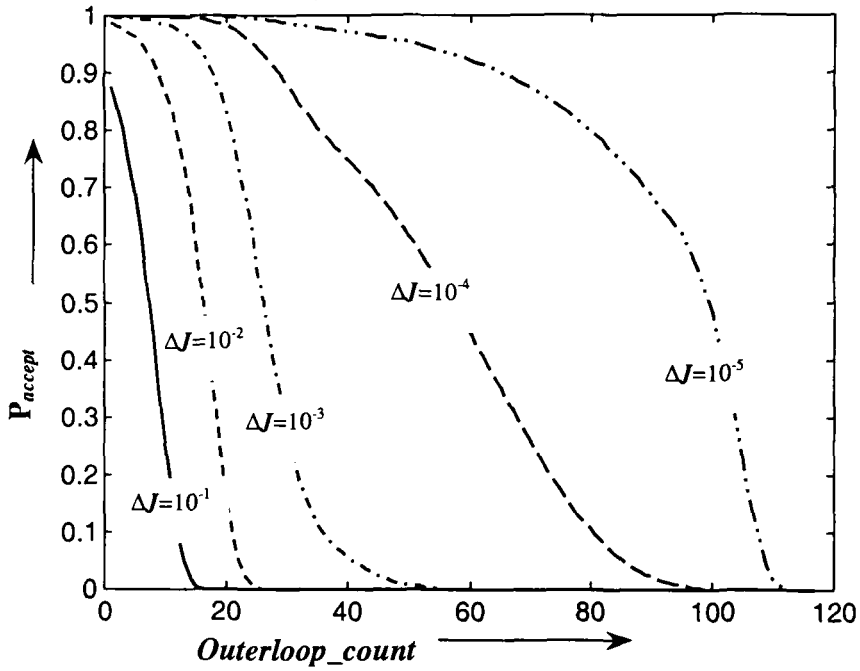


Fig. 5.7: Plot of $\mathbf{P}_{accept}$ vs outerloop_count for different values of ΔJ .

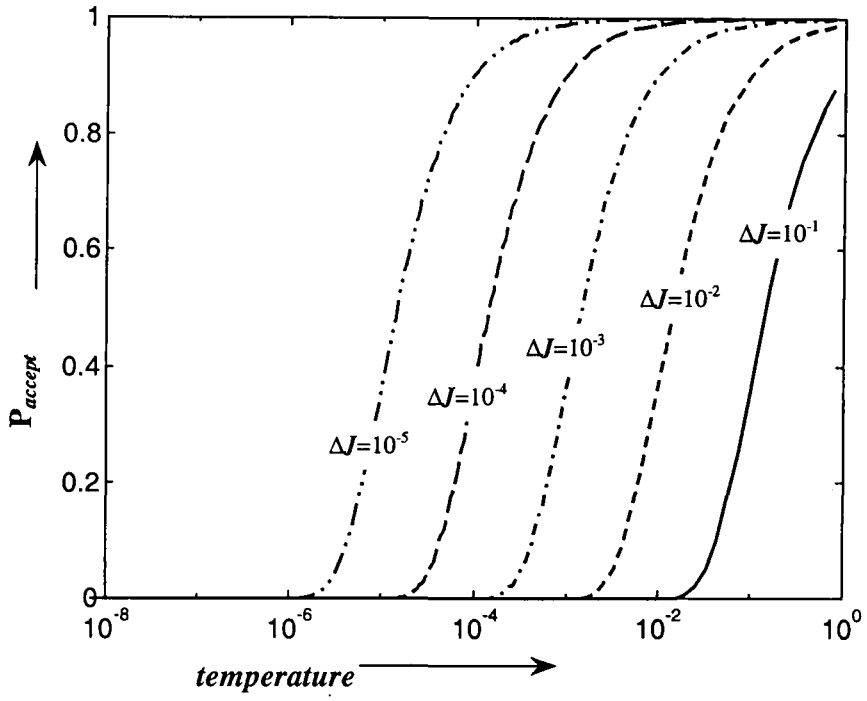


Fig. 5.8: Plot of $\mathbf{P}_{accept}$ vs temperature for different values of ΔJ .

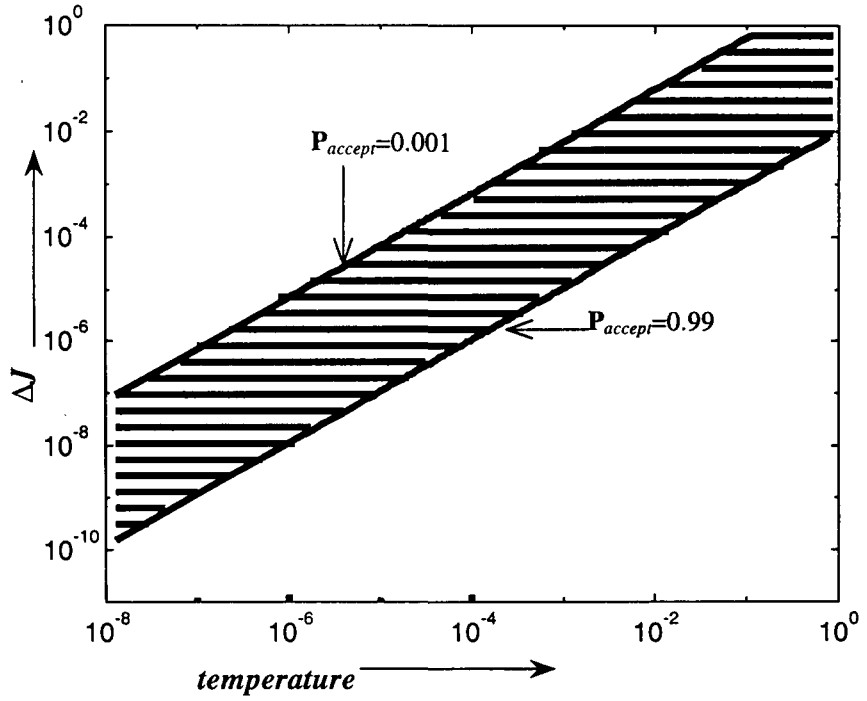


Fig. 5.9: Band of ΔJ at different temperature for $0.001 \leq \mathbf{P}_{accept} \leq 0.99$.

5-6 Simulation Results

To show the effect of learning algorithm, we present the best performance models for *Example 2.1* to *Example 2.4*. Since the models of *Example 2.1* to *Example 2.3* have achieved the target value of the objective function, we present these models with GD and LSE learning. The effects of SA hybrid and GA hybrid learning are shown on *Example 2.4*.

In this section, we make an exhaustive study by constructing basic rules of GFM followed by local learning using hybrids. The graphs for premise variable membership functions are analyzed. If they are flat, they are removed from the rules, thus simplifying the rules. These graphs thus depict various membership functions.

Example 2.1 Revisited

The initial class II GFM rules for data 1, obtained from the Modified Mountain clustering, are given below:

R^1 : if $y(t-1)$ is $A_1^1 \wedge y(t-2)$ is $A_2^1 \wedge u_2(t-4)$ is $A_3^1 \wedge u_1(t-3)$ is $A_4^1 \wedge u_1(t-1)$ is A_5^1 then $y(t)$ is $B^1\left(f^1(x), 2.02\right)$;

R^2 : if $y(t-1)$ is $A_1^2 \wedge y(t-2)$ is $A_2^2 \wedge u_2(t-4)$ is $A_3^2 \wedge u_1(t-3)$ is $A_4^2 \wedge u_1(t-1)$ is A_5^2 then $y(t)$ is $B^2\left(f^2(x), 2.02\right)$;

R^3 : if $y(t-1)$ is $A_1^3 \wedge y(t-2)$ is $A_2^3 \wedge u_2(t-4)$ is $A_3^3 \wedge u_1(t-3)$ is $A_4^3 \wedge u_1(t-1)$ is A_5^3 then $y(t)$ is $B^3\left(f^3(x), 2.02\right)$;

where,

$$f^1(x) = -0.91 + 0.80y(t-1) + 0.43y(t-2) + 2.02u_2(t-4) - 0.20u_1(t) - 1.89u_1(t-1)$$

$$f^2(x) = 1.03 + 0.19y(t-1) - 0.42y(t-2) + 0.10u_2(t-4) - 0.18u_1(t) + 2.04u_1(t-1)$$

$$f^3(x) = -2.83 + 0.57y(t-2) + 0.86u_2(t-4) + 4.48u_1(t) + 2.53u_1(t-1)$$

the premise variable membership functions A_1^1 to A_1^3, A_2^1 to A_2^3, A_3^1 to A_3^3, A_4^1 to A_4^3 and A_5^1 to A_5^3 are shown in Fig.5.10(a-e) respectively. We have used GD and

LSE methods for learning the GFM. Final fuzzy rules corresponding to the learned GFM are listed below:

R^{1f} : if $y(t-1)$ is $A_1^{1f} \wedge y(t-2)$ is $A_2^{1f} \wedge u_2(t-4)$ is $A_3^{1f} \wedge u_1(t)$ is $A_4^{1f} \wedge u_1(t-1)$ is A_5^{1f} then $y(t)$ is $B^{1f}\left(f^{1f}(x), 1.96\right)$;

R^{2f} : if $y(t-1)$ is $A_1^{2f} \wedge y(t-2)$ is $A_2^{2f} \wedge u_2(t-4)$ is $A_3^{2f} \wedge u_1(t)$ is $A_4^{2f} \wedge u_1(t-1)$ is A_5^{2f} then $y(t)$ is $B^{2f}\left(f^{2f}(x), 2.02\right)$;

R^{3f} : if $y(t-1)$ is $A_1^{3f} \wedge y(t-2)$ is $A_2^{3f} \wedge u_2(t-4)$ is $A_3^{3f} \wedge u_1(t)$ is $A_4^{3f} \wedge u_1(t-1)$ is A_5^{3f} then $y(t)$ is $B^{3f}\left(f^{3f}(x), 2.05\right)$;

where,

$$f^{1f}(x) = -0.69 + 1.00y(t-1) + 0.18y(t-2) + 1.00u_2(t-4) + 0.88u_1(t) - 1.32u_1(t-1)$$
$$f^{2f}(x) = 1.27 + 0.18y(t-1) - 0.37y(t-2) + 1.00u_2(t-4) - 1.70u_1(t) + 1.28u_1(t-1)$$
$$f^{3f}(x) = -2.92 - 0.17y(t-1) + 0.81y(t-2) + 1.00u_2(t-4) + 4.13u_1(t) + 2.30u_1(t-1)$$

The learned membership functions of premise variables A_1^{1f} to A_1^{3f}, A_2^{1f} to A_2^{3f}, A_3^{1f} to A_3^{3f}, A_4^{1f} to A_4^{3f} and A_5^{1f} to A_5^{3f} are shown in Fig.5.11(a-e) respectively. There is a little effect of GD learning on the premise membership function but we can see that there is a small significant change in the values of parameter v_k. Effect of LSE learning is quite significant. The plots of actual output y, model output y^o and model error are shown in Fig.4.7.

The initial class II GFM rules for data 2, obtained from the Modified Mountain clustering, are given below:

R^1 : if $y(t-1)$ is $A_1^1 \wedge y(t-2)$ is $A_2^1 \wedge u_2(t-4)$ is $A_3^1 \wedge u_1(t-3)$ is $A_4^1 \wedge u_1(t-1)$ is A_5^1 then $y(t)$ is $B^1\left(f^1(x), 2.30\right)$;

R^2 : if $y(t-1)$ is $A_1^2 \wedge y(t-2)$ is $A_2^2 \wedge u_2(t-4)$ is $A_3^2 \wedge u_1(t-3)$ is $A_4^2 \wedge u_1(t-1)$ is A_5^2 then $y(t)$ is $B^2\left(f^2(x), 2.30\right)$;

R^3 : if $y(t-1)$ is $A_1^3 \wedge y(t-2)$ is $A_2^3 \wedge u_2(t-4)$ is $A_3^3 \wedge u_1(t-3)$ is $A_4^3 \wedge u_1(t-1)$ is A_5^3 then $y(t)$ is $B^3\left(f^3(x), 2.30\right)$;

where,

$$f^1(x) = -4.44 + 1.07y(t-1) + 1.31y(t-2) + 1.28u_2(t-4) + 2.30u_1(t) + 0.47u_1(t-1)$$
$$f^2(x) = 3.52 \qquad -1.56y(t-2) + 1.40u_2(t-4) - 2.02u_1(t) + 1.02u_1(t-1)$$

$$f^3(x) = -1.25 - 0.22y(t-2) + 0.81y(t-2) + 0.14u_2(t-4) + 3.71u_1(t) + 0.91u_1(t-1)$$

The premise variable membership functions A_1^1 to A_1^3, A_2^1 to A_2^3, $A_3^1 - A_3^3$, A_4^1 to A_4^3 and A_5^1 to A_5^3 are shown in Fig.5.12(a-e) respectively. Final fuzzy rules corresponding to the learned GFM are listed below:

R^{1f} : if $y(t-1)$ is $A_1^{1f} \wedge y(t-2)$ is $A_2^{1f} \wedge u_2(t-4)$ is $A_3^{1f} \wedge u_1(t)$ is $A_4^{1f} \wedge u_1(t-1)$ is A_5^{1f} then $y(t)$ is $B^{1f}\left(f^{1f}(x), 2.33\right)$;

R^{2f} : if $y(t-1)$ is $A_1^{2f} \wedge y(t-2)$ is $A_2^{2f} \wedge u_2(t-4)$ is $A_3^{2f} \wedge u_1(t)$ is $A_4^{2f} \wedge u_1(t-1)$ is A_5^{2f} then $y(t)$ is $B^{2f}\left(f^{2f}(x), 2.12\right)$;

R^{3f} : if $y(t-1)$ is $A_1^{3f} \wedge y(t-2)$ is $A_2^{3f} \wedge u_2(t-4)$ is $A_3^{3f} \wedge u_1(t)$ is $A_4^{3f} \wedge u_1(t-1)$ is A_5^{3f} then $y(t)$ is $B^{3f}\left(f^{3f}(x), 2.43\right)$;

where,

$$f^{1f}(x) = -4.26 + 2.21y(t-1) + 1.69y(t-2) + 0.89u_2(t-4) + 2.16u_1(t) - 0.08u_1(t-1)$$

$$f^{2f}(x) = 2.45 - 0.93y(t-1) - 1.12y(t-2) + 1.18u_2(t-4) - 0.81u_1(t) + 1.57u_1(t-1)$$

$$f^{3f}(x) = 0.02 - 0.17y(t-1) - 0.18y(t-2) + 0.54u_2(t-4) + 2.41u_1(t) - 1.48u_1(t-1)$$

The learned membership functions of premise variables A_1^{1f} to A_1^{3f}, A_2^{1f} to A_2^{3f}, A_3^{1f} to A_3^{3f}, A_4^{1f} to A_4^{3f} and A_5^{1f} to A_5^{3f} are shown in Fig.5.13(a-e) respectively. There is a significant effect of GD learning on the premise membership functions A_2^{2f} and A_3^{3f}, and they remain at the value of "1" through their supports. Owing to this, they don't contribute to the firing strength in the respective rules, therefore $y(t-2)$ and $u_2(t-4)$ can be removed from R^{2f} and R^{3f}, respectively, and the fuzzy rules can be rewritten as follows:

R^{1f} : if $y(t-1)$ is $A_1^{1f} \wedge y(t-2)$ is $A_2^{1f} \wedge u_2(t-4)$ is $A_3^{1f} \wedge u_1(t)$ is $A_4^{1f} \wedge u_1(t-1)$ is A_5^{1f} then $y(t)$ is $B^{1f}\left(f^{1f}(x), 2.33\right)$;

R^{2f} : if $y(t-1)$ is $A_1^{2f} \wedge$ $u_2(t-4)$ is $A_3^{2f} \wedge u_1(t)$ is $A_4^{2f} \wedge u_1(t-1)$ is A_5^{2f} then $y(t)$ is $B^{2f}\left(f^{2f}(x), 2.12\right)$;

R^{3f} : if $y(t-1)$ is $A_1^{3f} \wedge y(t-2)$ is $A_2^{3f} \wedge$ $u_1(t)$ is $A_4^{3f} \wedge u_1(t-1)$ is A_5^{3f} then $y(t)$ is $B^{3f}\left(f^{3f}(x), 2.43\right)$;

Effect of LSE learning is quite significant. Figure 4.11 shows the actual output y, model output y^o and model error for data 2.

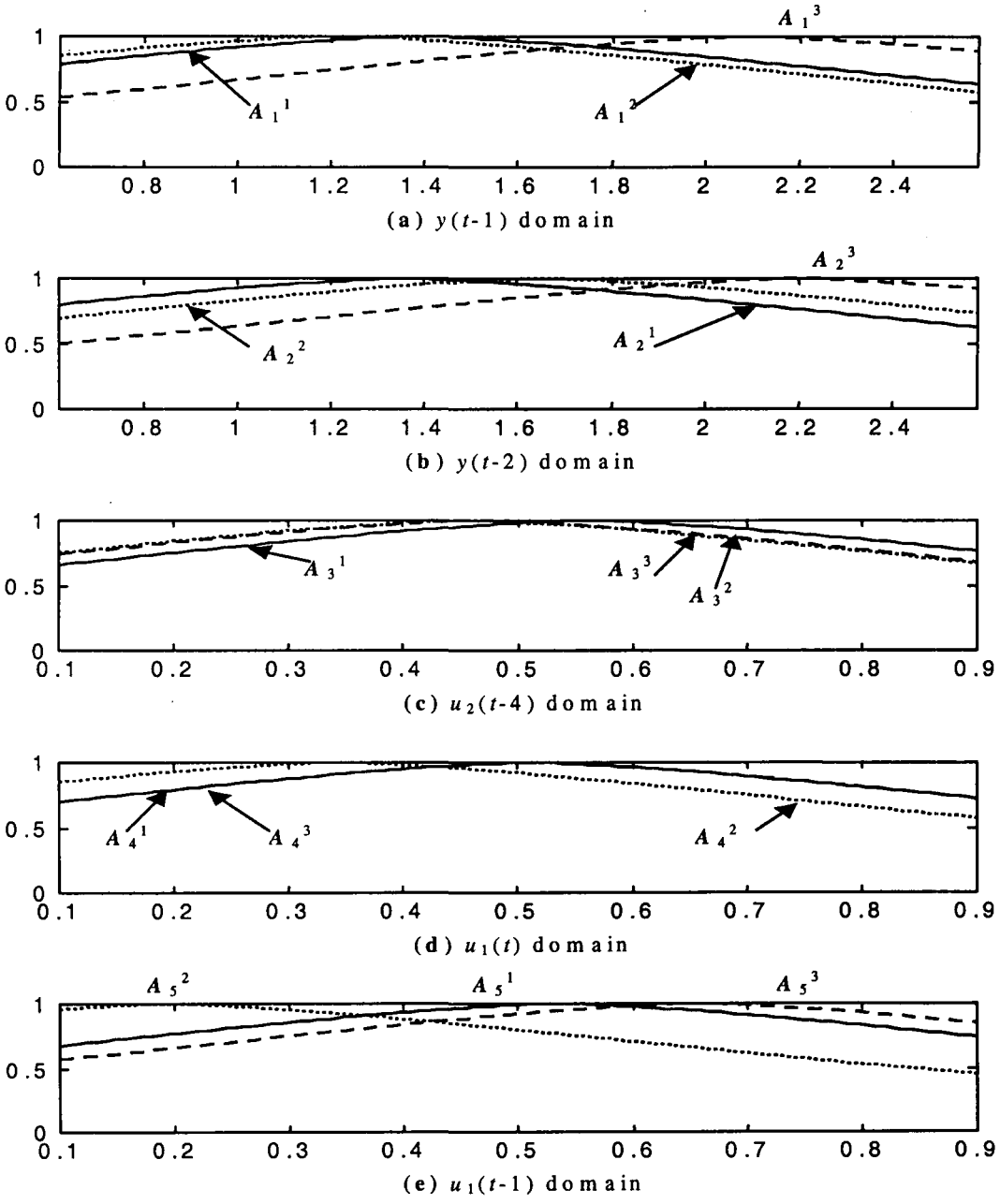


Fig. 5.10: Initial membership functions for class II GFM rules, obtained from the Modified Mountain clustering, for data 1 of *Example 2.1.*

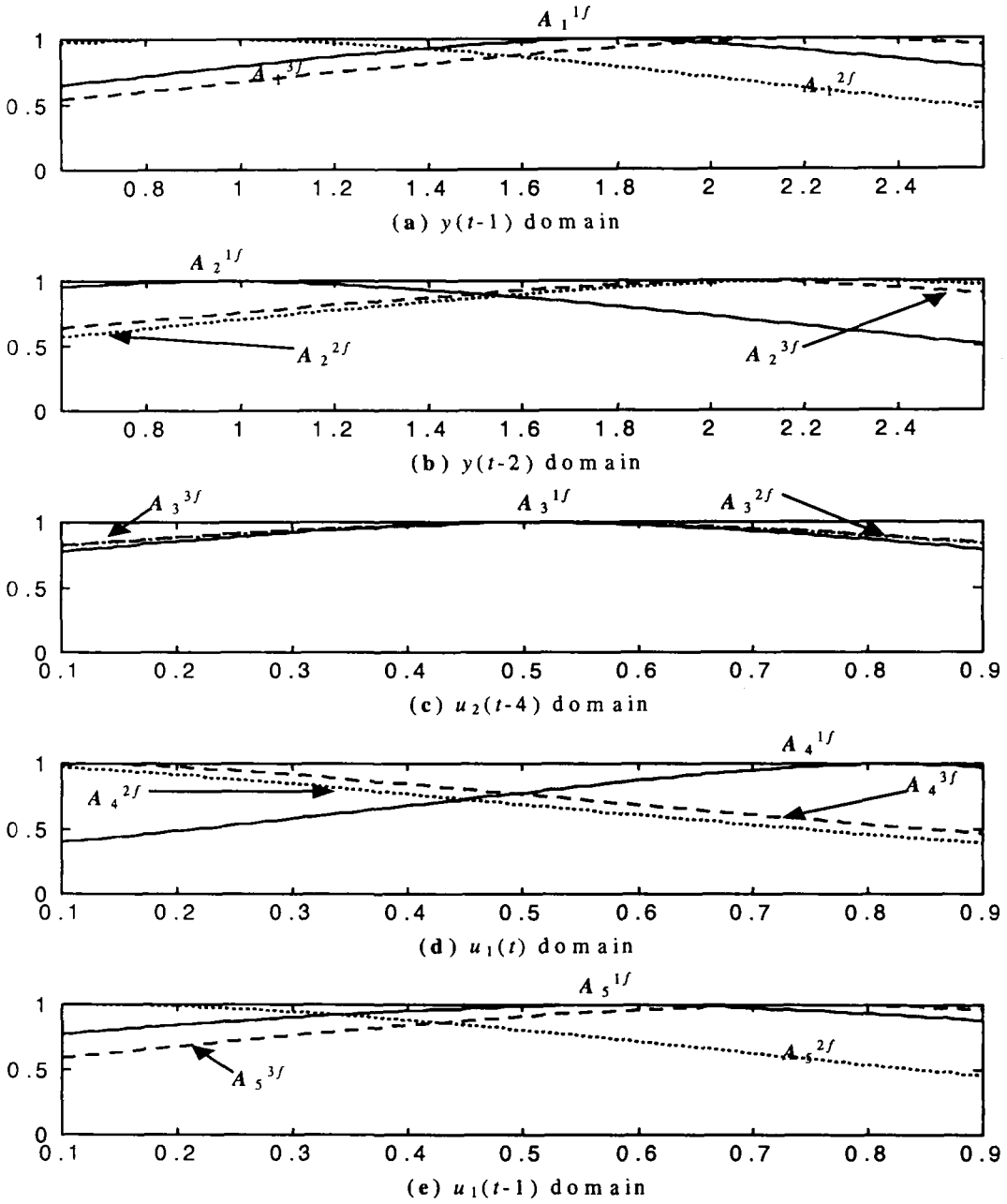


Fig. 5.11: Final membership functions for class II GFM rules, learned by GD, for data 1 of *Example 2.1.*

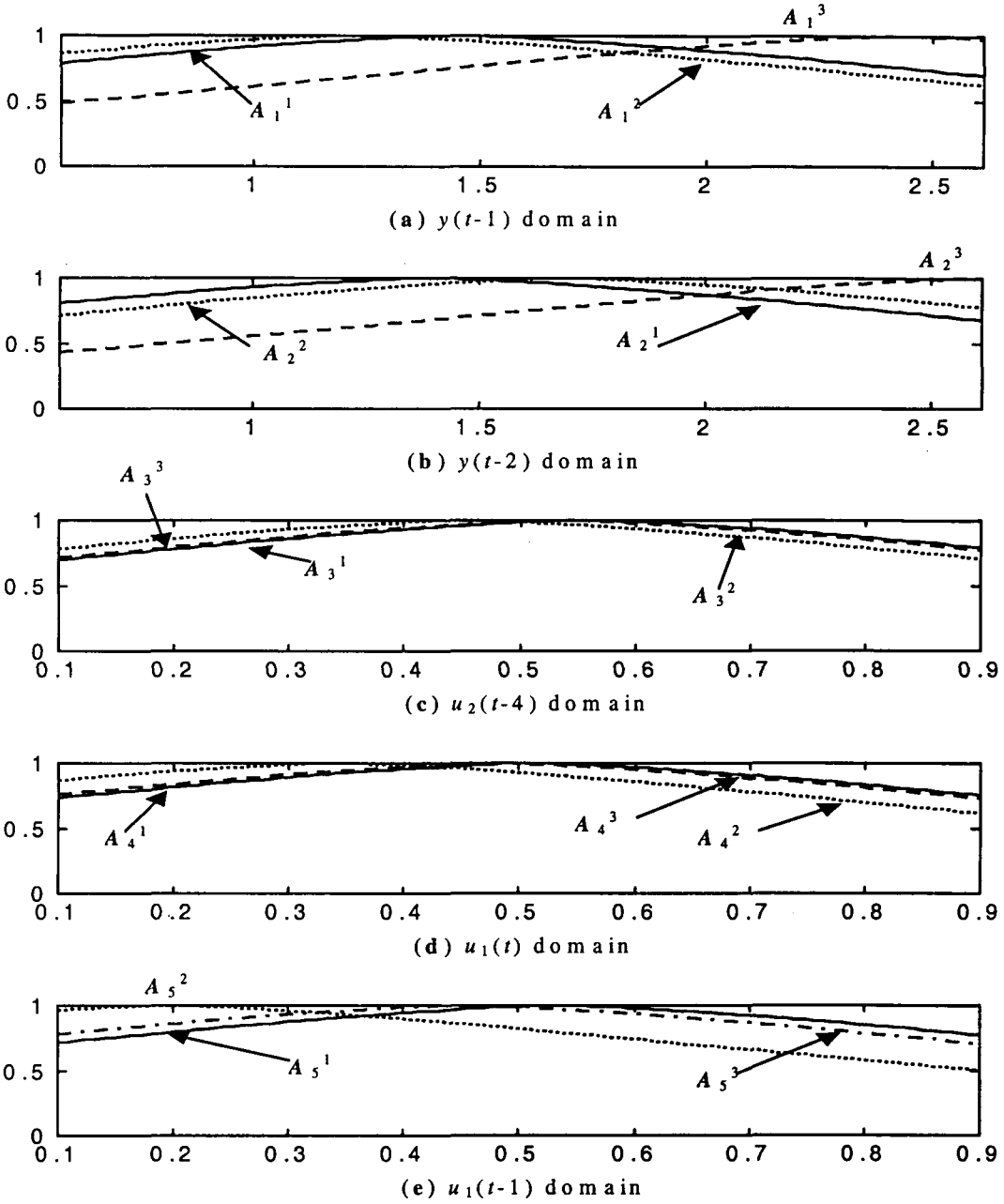


Fig. 5.12: Initial membership functions for class II GFM rules, obtained from the Modified Mountain clustering, for data 2 of *Example 2.1*.

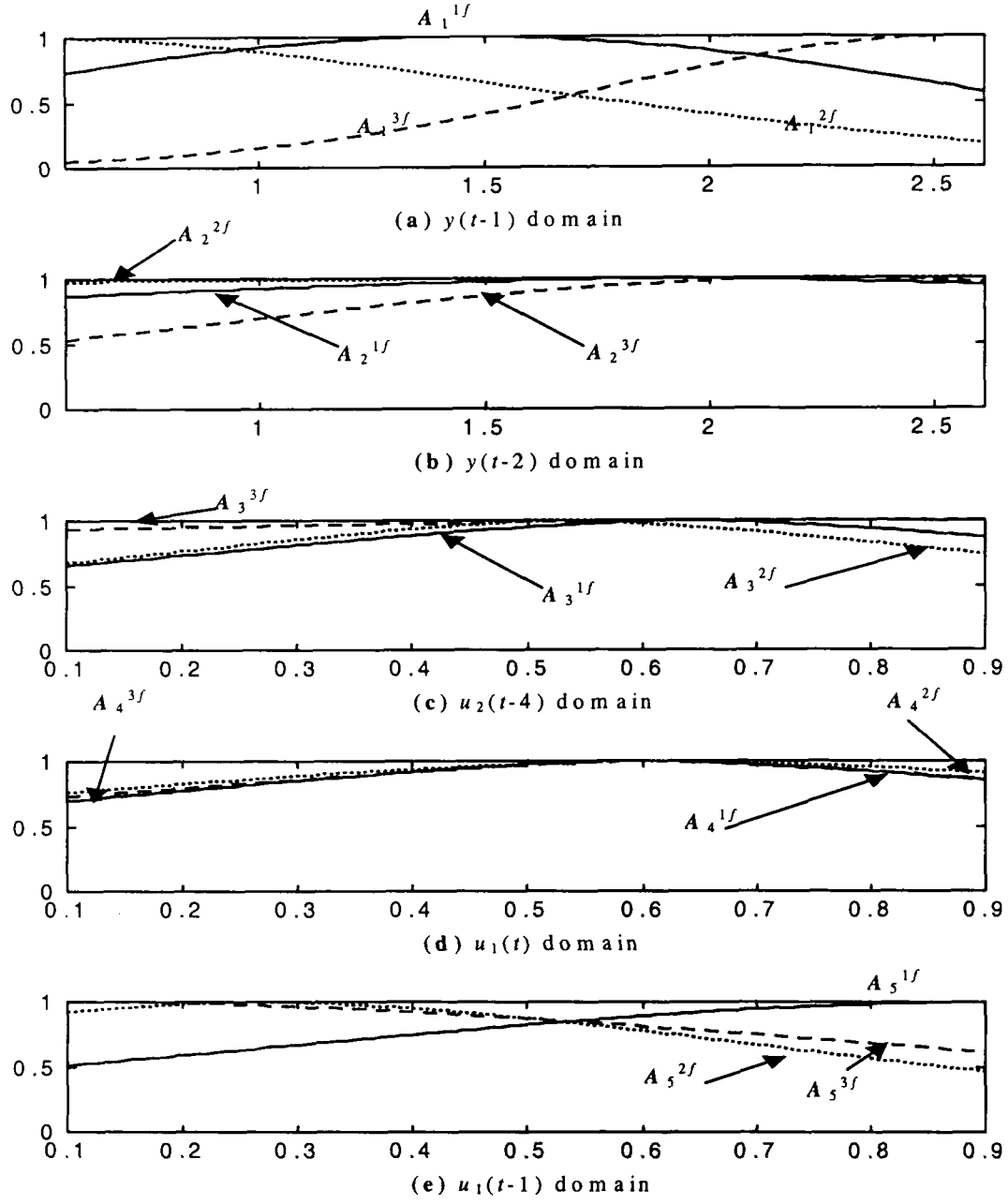


Fig. 5.13: Final membership functions for class II GFM rules, learned by GD, for data 2 of *Example 2.1.*

It may be observed that in Figs. 5.10,& 5.12 the initial membership function obtained from modified mountain clustering are almost overlapping. This indicates that the corresponding fuzzy sets in the rules have the similar sense, but after learning they get change as shown in Figs. 5.11,& 5.13. Those initial membership functions, which are overlapping, may not over lap after learning. However some other final membership function overlap after learning. This indicates the changing nature of membership function, reflecting the dynamic behaviour of learning ability of GFM.

Example 2.2 Revisited

The initial class II GFM rules, obtained from the Modified Mountain clustering, are given below:

R^1 : if $u(t-5)$ is $A_1^1 \wedge u(t-4)$ is $A_2^1 \wedge y(t-1)$ is $A_3^1 \wedge u(t-3)$ is $A_4^1 \wedge y(t-2)$ is A_5^1 then $y(t)$ is $B^1\left(f^1(x), 17.88\right)$;

R^2 : if $u(t-5)$ is $A_1^2 \wedge u(t-4)$ is $A_2^2 \wedge y(t-1)$ is $A_3^2 \wedge u(t-3)$ is $A_4^2 \wedge y(t-2)$ is A_5^2 then $y(t)$ is $B^2\left(f^2(x), 17.88\right)$;

R^3 : if $u(t-5)$ is $A_1^3 \wedge u(t-4)$ is $A_2^3 \wedge y(t-1)$ is $A_3^3 \wedge u(t-3)$ is $A_4^3 \wedge y(t-2)$ is A_5^3 then $y(t)$ is $B^3\left(f^2(x), 17.88\right)$;

where,

$$f^1(x) = -6.02 - 0.48u(t-5) - 0.75u(t-4) + 2.30y(t-1) + 0.44u(t-3) - 1.20y(t-2)$$

$$f^2(x) = 10.67 + 0.33u(t-5) + 0.03u(t-4) + 0.63y(t-1) - 1.27u(t-3) + 0.13y(t-2)$$

$$f^3(x) = 28.78 - 0.20u(t-5) + 0.36u(t-4) - 0.24y(t-1) - 2.07u(t-3) + 0.75y(t-2)$$

The premise variable membership functions A_1^1 to A_1^3, A_2^1 to A_2^3, A_3^1 to A_3^3, A_4^1 to A_4^3 and A_5^1 to A_5^3 are shown in Fig.5.14(a-e) respectively. Final fuzzy rules corresponding to the learned GFM are listed below:

R^{1f} : if $u(t-5)$ is $A_1^{1f} \wedge u(t-4)$ is $A_2^{1f} \wedge y(t-1)$ is $A_3^{1f} \wedge u(t-3)$ is $A_4^{1f} \wedge y(t-2)$ is A_5^{1f} then $y(t)$ is $B^{1f}\left(f^{1f}(x), 17.87\right)$;

R^{2f} : if $u(t-5)$ is $A_1^{2f} \wedge u(t-4)$ is $A_2^{2f} \wedge y(t-1)$ is $A_3^{2f} \wedge u(t-3)$ is $A_4^{2f} \wedge y(t-2)$ is A_5^{2f} then $y(t)$ is $B^{2f}\left(f^{2f}(x), 17.86\right)$;

R^{3f} : if $u(t-5)$ is $A_1^{3f} \wedge u(t-4)$ is $A_2^{3f} \wedge y(t-1)$ is $A_3^{3f} \wedge u(t-3)$ is $A_4^{3f} \wedge y(t-2)$ is A_5^{3f} then $y(t)$ is $B^{3f}\left(f^{3f}(x), 17.88\right)$;

where,

$$f^{1f}(x) = 14.21 + 0.04u(t-5) - 0.07u(t-4) + 0.81y(t-1) - 0.92u(t-3) - 0.08y(t-2)$$

$$f^{2f}(x) = 11.05 + 0.17u(t-5) - 0.55u(t-4) + 1.08y(t-1) - 0.32u(t-3) - 0.29y(t-2)$$

$$f^{3f}(x) = 12.36 - 0.09u(t-5) + 0.21u(t-4) + 1.16y(t-1) - 0.75u(t-3) - 0.39y(t-2)$$

The learned membership functions of the premise variables A_1^{1f} to A_1^{3f}, A_2^{1f} to A_2^{3f}, A_3^{1f} to A_3^{3f}, A_4^{1f} to A_4^{3f} and A_5^{1f} to A_5^{3f} are shown in Fig.5.15(a-e)

respectively. There is a significant effect of GD learning on the premise membership functions A_1^{1f}, A_1^{3f}, A_2^{1f} and A_4^{2f}, and they remain at the value of "1" through their supports. Owing to this they don't contribute to the firing strength in the respective rules, therefore $u(t-5)$ and $u(t-4)$ can be removed from R^{1f}, $y(t-1)$ from R^{2f} and $u(t-5)$ from R^{3f}, and the fuzzy rules can be rewritten as follows:

R^{1f} : if $y(t-1)$ is $A_3^{1f} \wedge u(t-3)$ is $A_4^{1f} \wedge y(t-2)$ is A_5^{1f} then $y(t)$ is $B^{1f}\left(f^{1f}(x), 17.87\right)$;

R^{2f} : if $u(t-5)$ is $A_1^{2f} \wedge u(t-4)$ is $A_2^{2f} \wedge y(t-1)$ is $A_3^{2f} \wedge y(t-2)$ is A_5^{2f} then $y(t)$ is $B^{2f}\left(f^{2f}(x), 17.86\right)$;

R^{3f} : if $u(t-4)$ is $A_2^{3f} \wedge y(t-1)$ is $A_3^{3f} \wedge u(t-3)$ is $A_4^{3f} \wedge y(t-2)$ is A_5^{3f} then $y(t)$ is $B^{3f}\left(f^{3f}(x), 17.88\right)$;

Figure 4.15 shows the actual output y, model output y^o and model error.

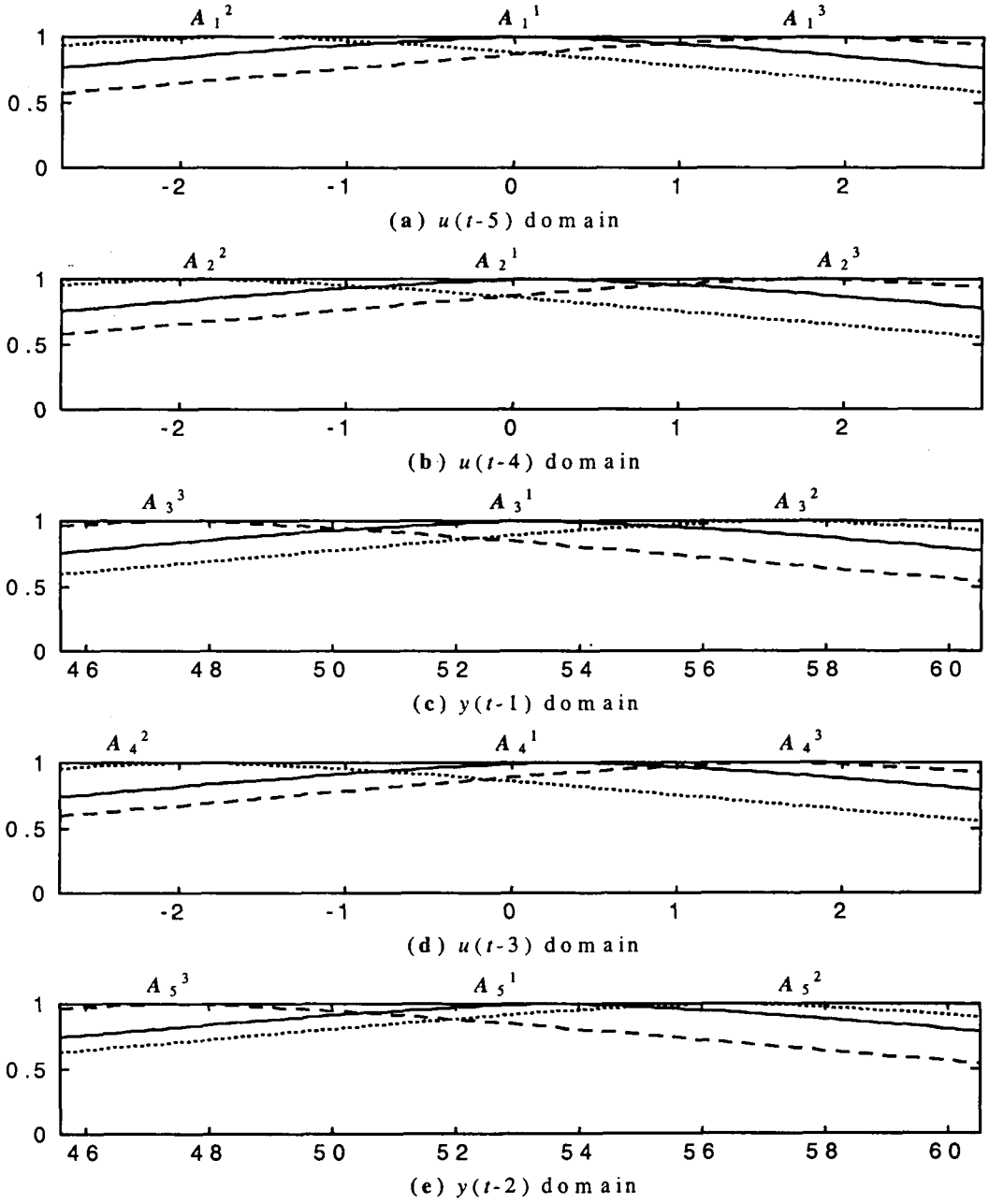


Fig. 5.14: Initial membership functions for class II GFM rules, obtained from the Modified Mountain clustering, of *Example 2.2*.

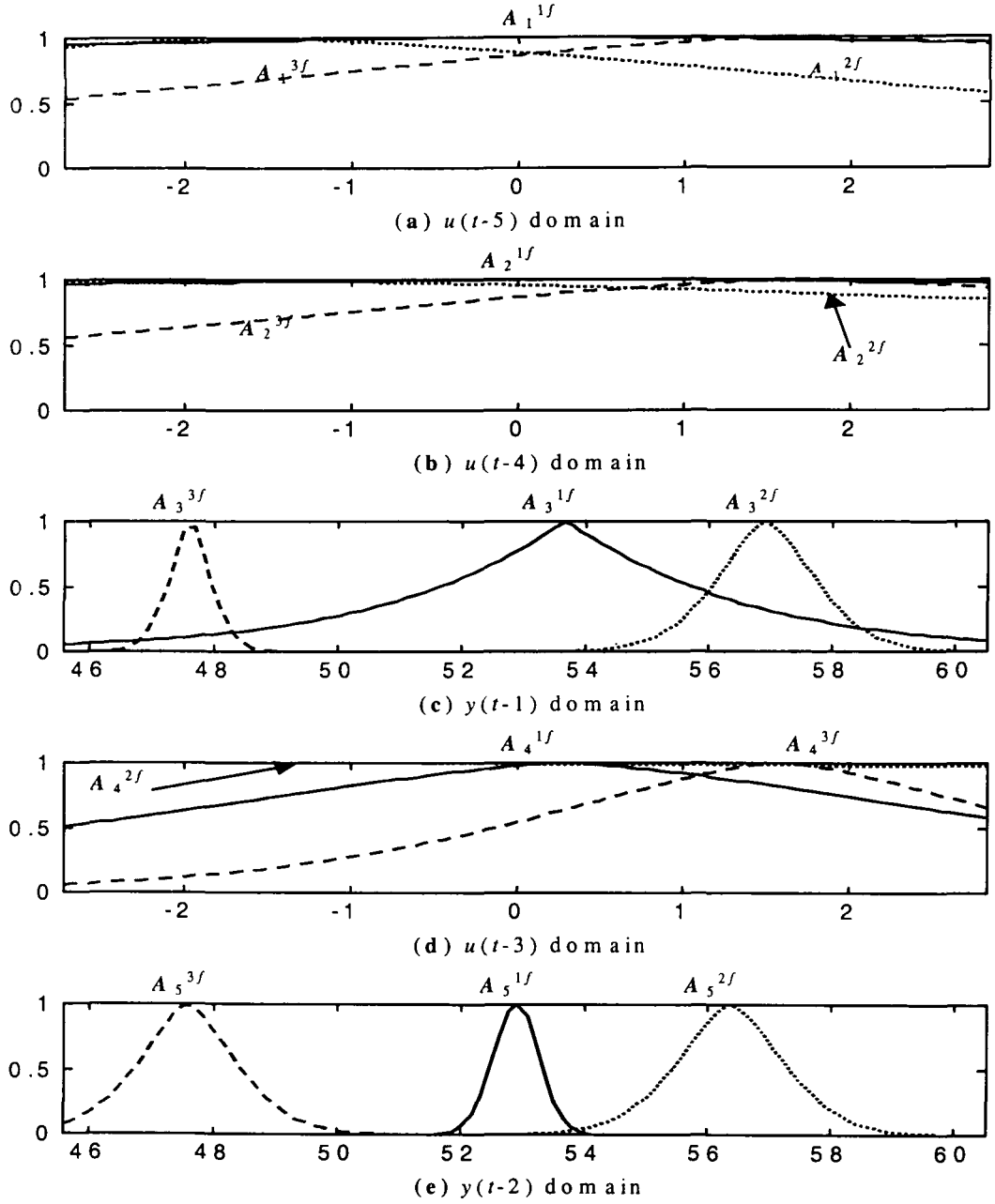


Fig. 5.15: Final membership functions for class II GFM rules, learned by GD, of *Example 2.2*.

Example 2.3 Revisited

The initial class II GFM rules, obtained from the Modified Mountain clustering, are given below:

$$R^1: \text{if } u_3 \text{ is } A_1^1 \wedge u_1 \text{ is } A_2^1 \quad \text{then } y \text{ is } B^1\left(f^1(x), 2.36\right);$$

$$R^2: \text{if } u_3 \text{ is } A_1^2 \wedge u_1 \text{ is } A_2^2 \quad \text{then } y \text{ is } B^2\left(f^2(x), 2.36\right);$$

$$R^3: \text{if } u_3 \text{ is } A_1^3 \wedge u_1 \text{ is } A_2^3 \quad \text{then } y \text{ is } B^3\left(f^3(x), 2.36\right);$$

$$R^4: \text{if } u_3 \text{ is } A_1^4 \wedge u_1 \text{ is } A_2^4 \quad \text{then } y \text{ is } B^4\left(f^4(x), 2.36\right);$$

$$R^5: \text{if } u_3 \text{ is } A_1^5 \wedge u_1 \text{ is } A_2^5 \quad \text{then } y \text{ is } B^5\left(f^5(x), 2.36\right);$$

where,

$$f^1(x) = -281150 + 27.34u_3 + 25809u_1$$

$$f^2(x) = 181710 - 3.25u_3 - 29717u_1$$

$$f^3(x) = 9020 + 2.23u_3 - 1798u_1$$

$$f^4(x) = 8590 + 0.27u_3 - 1156u_1$$

$$f^5(x) = 149250 - 19.89u_3 - 8790u_1$$

The premise variable membership functions A_1^1 to A_1^5 and A_2^1 to A_2^5 are shown in Fig.5.16(a-b) respectively. Final fuzzy rules corresponding to the learned GFM are listed below:

$$R^{1f}: \text{if } u_3 \text{ is } A_1^{1f} \wedge u_1 \text{ is } A_2^{1f} \quad \text{then } y \text{ is } B^{1f}\left(f^{1f}(x), 2.36\right);$$

$$R^{2f}: \text{if } u_3 \text{ is } A_1^{2f} \wedge u_1 \text{ is } A_2^{2f} \quad \text{then } y \text{ is } B^{2f}\left(f^{2f}(x), 2.36\right);$$

$$R^{3f}: \text{if } u_3 \text{ is } A_1^{3f} \wedge u_1 \text{ is } A_2^{3f} \quad \text{then } y \text{ is } B^{3f}\left(f^{3f}(x), 2.36\right);$$

$$R^{4f}: \text{if } u_3 \text{ is } A_1^{4f} \wedge u_1 \text{ is } A_2^{4f} \quad \text{then } y \text{ is } B^{4f}\left(f^{4f}(x), 2.36\right);$$

$$R^{5f}: \text{if } u_3 \text{ is } A_1^{5f} \wedge u_1 \text{ is } A_2^{5f} \quad \text{then } y \text{ is } B^{5f}\left(f^{5f}(x), 2.36\right);$$

where,

$$f^{1f}(\boldsymbol{x}) = -213750 + 44.56u_3 - 14975u_1$$
$$f^{2f}(\boldsymbol{x}) = -6650 + 10.79u_3 - 5336u_1$$
$$f^{3f}(\boldsymbol{x}) = 3380 + 1.01u_3 - 538u_1$$
$$f^{4f}(\boldsymbol{x}) = 4780 + 0.40u_3 - 657u_1$$
$$f^{5f}(\boldsymbol{x}) = -191980 - 32.71u_3 - 2012u_1$$

The learned membership functions of the premise variables A_1^{1f} to A_1^{5f} and A_2^{1f} to A_2^{5f} are shown in Fig.5.17(a-b) respectively. Figure 4.19 shows the actual output y, model output y^o and model error.

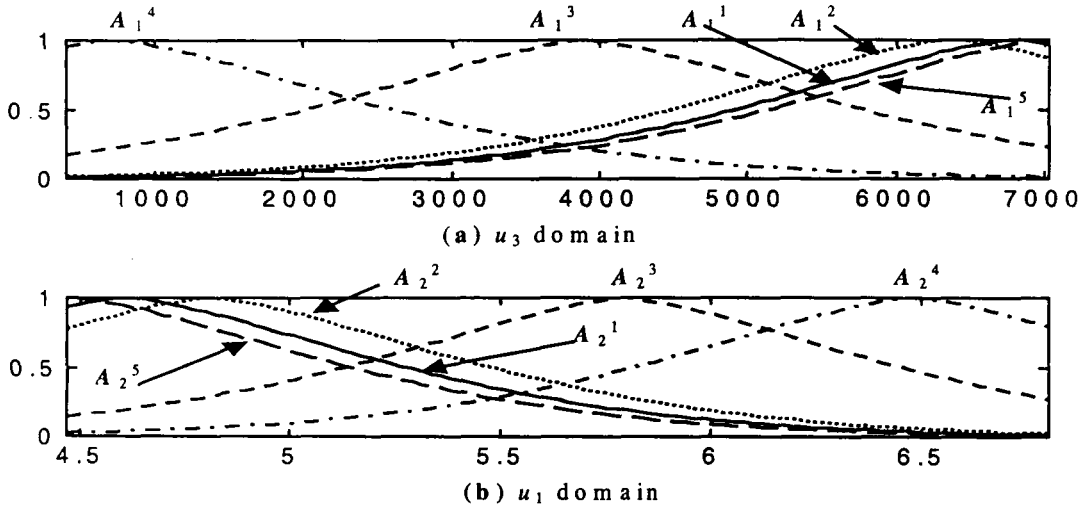


Fig. 5.16: Initial membership functions for class II GFM rules, obtained from the Modified Mountain clustering, of *Example 2.3*.

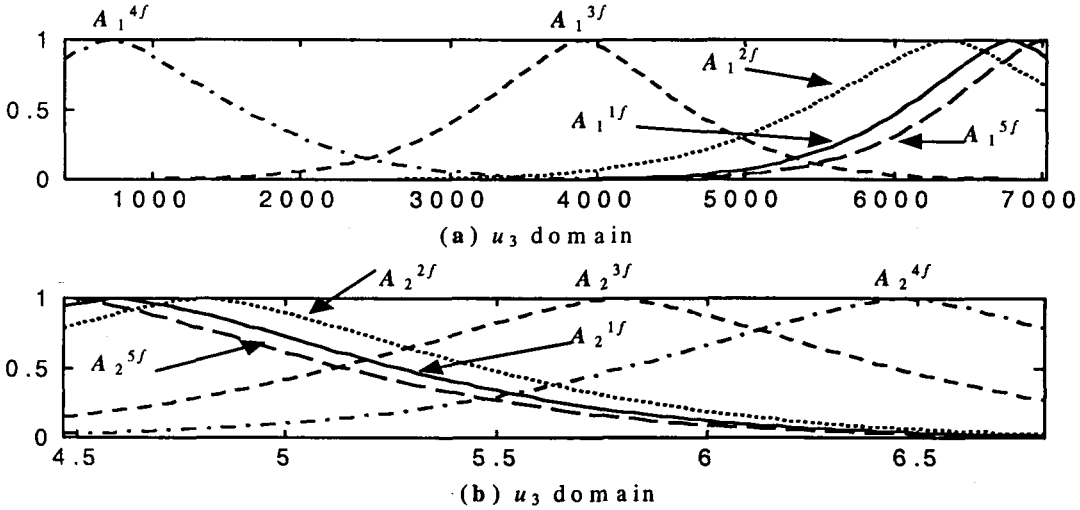


Fig. 5.17: Final membership functions for class II GFM rules, learned by GD, of *Example 2.3.*

Example 2.4 Revisited

Since the performance of models for a stock market data have not reached the target value, we apply the SA hybrid scheme and the GA hybrid scheme for model learning. First we shall present the results obtained using the SA hybrid algorithm, followed by the GA hybrid algorithm.

For SA hybrid scheme, we take the initial model from the fuzzy curve method. The initial class II GFM rules are given below:

R^1 : if x_4 is $A_1^1 \wedge x_2$ is $A_2^1 \wedge x_5$ is $A_3^1 \wedge x_8$ is $A_4^1 \wedge x_{10}$ is $A_5^1 \wedge x_1$ is A_6^1 then y is $B^1\left(f^1(\boldsymbol{x}), 20.70\right)$;

R^2 : if x_4 is $A_1^2 \wedge x_2$ is $A_2^2 \wedge x_5$ is $A_3^2 \wedge x_8$ is $A_4^2 \wedge x_{10}$ is $A_5^2 \wedge x_1$ is A_6^2 then y is $B^2\left(f^2(\boldsymbol{x}), 20.70\right)$;

R^3 : if x_4 is $A_1^3 \wedge x_2$ is $A_2^3 \wedge x_5$ is $A_3^3 \wedge x_8$ is $A_4^3 \wedge x_{10}$ is $A_5^3 \wedge x_1$ is A_6^3 then y is $B^3\left(f^3(\boldsymbol{x}), 20.70\right)$;

R^4 : if x_4 is $A_1^4 \wedge x_2$ is $A_2^4 \wedge x_5$ is $A_3^4 \wedge x_8$ is $A_4^4 \wedge x_{10}$ is $A_5^4 \wedge x_1$ is A_6^4 then y is $B^4\left(f^4(\boldsymbol{x}), 20.70\right)$;

where,

$$f^1(\boldsymbol{x}) = -84.03 + 0.49x_4 + 20.13x_2 - 19.26x_5 + 1.81x_8 + 0.11x_{10} + 193.95x_1$$

$$f^2(\boldsymbol{x}) = 29.78 - 2.06x_4 + 53.30x_2 + 7.90x_5 - 2.13x_8 - 1.08x_{10} - 113.04x_1$$

$$f^3(x) = 22.74 - 8.93x_4 + 793.69x_2 - 32.14x_5 + 3.85x_8 + 4.08x_{10} - 95.72x_1$$

$$f^4(x) = -93.32 + 2.89x_4 - 748.87x_2 - 7.99x_5 - 1.56x_8 - 3.90x_{10} - 173.27x_1$$

The premise variable membership functions A_1^1 to A_1^4, A_2^1 to A_2^4, A_3^1 to A_3^4, A_4^1 to A_4^4 A_5^1 to A_5^4 and A_6^1 to A_6^4 are shown in Fig.5.19(a-f) respectively. We have first used SA hybrid with both GD and LSE methods for learning the GFM.

Figure 5.18 shows the learning pattern of SA. The moves with very high value of ΔJ are accepted initially, i.e., at high temperature, while at lower temperature the solution converges to one of the basins. In the middle temperature zone, the solution remains on flat plateau in most of the moves. Sometimes it goes into the basins, but soon it moves out from there.

Final fuzzy rules corresponding to the GFM learned by the SA hybrid scheme are listed below:

R_s^{1f} : if x_4 is A_{1s}^{1f} $\wedge x_2$ is A_{2s}^{1f} $\wedge x_5$ is A_{3s}^{1f} $\wedge x_8$ is A_{4s}^{1f} $\wedge x_{10}$ is A_{5s}^{1f} $\wedge x_1$ is A_{6s}^{1f} then y is $B_s^{1f}\left(f_s^{1f}(x), 20.62\right)$;

R_s^{2f} : if x_4 is A_{1s}^{2f} $\wedge x_2$ is A_{2s}^{2f} $\wedge x_5$ is A_{3s}^{2f} $\wedge x_8$ is A_{4s}^{2f} $\wedge x_{10}$ is A_{5s}^{2f} $\wedge x_1$ is A_{6s}^{2f} then y is $B_s^{2f}\left(f_s^{2f}(x), 20.61\right)$;

R_s^{3f} : if x_4 is A_{1s}^{3f} $\wedge x_2$ is A_{2s}^{3f} $\wedge x_5$ is A_{3s}^{3f} $\wedge x_8$ is A_{4s}^{3f} $\wedge x_{10}$ is A_{5s}^{3f} $\wedge x_1$ is A_{6s}^{3f} then y is $B_s^{3f}\left(f_s^{3f}(x), 20.61\right)$;

R_s^{4f} : if x_4 is A_{1s}^{4f} $\wedge x_2$ is A_{2s}^{4f} $\wedge x_5$ is A_{3s}^{4f} $\wedge x_8$ is A_{4s}^{4f} $\wedge x_{10}$ is A_{5s}^{4f} $\wedge x_1$ is A_{6s}^{4f} then y is $B_s^{4f}\left(f_s^{4f}(x), 20.62\right)$;

where,

$$f_s^{1f}(x) = 47.40 - 1.50x_4 + 10.30x_2 - 15.10x_5 - 0.20x_8 + 1.60x_{10} - 34.80x_1$$

$$f_s^{2f}(x) = 61.7 + 2.30x_4 - 418.50x_2 + 22.60x_5 - 1.20x_8 - 2.20x_{10} - 45.00x_1$$

$$f_s^{3f}(x) = -171.00 - 37.10x_4 + 6872.10x_2 + 34.50x_5 - 10.40x_8 + 8.40x_{10} - 507.30x_1$$

$$f_s^{4f}(x) = -66.70 - 5.40x_4 + 456.50x_2 - 23.70x_5 + 1.30x_8 + 2.20x_{10} - 7.50x_1$$

The learned membership functions of the premise variables A_{1s}^{1f} to A_{1s}^{4f}, A_{2s}^{1f} to A_{2s}^{4f}, A_{3s}^{1f} to A_{3s}^{4f}, A_{4s}^{1f} to A_{4s}^{4f} A_{5s}^{1f} to A_{5s}^{4f} and A_{6s}^{1f} to A_{6s}^{4f} are shown in Fig.5.20(a-f) respectively. The premise membership functions A_{1s}^{2f}, A_{3s}^{2f}, A_{3s}^{4f}, A_{4s}^{1f}, A_{4s}^{4f}, A_{5s}^{1f}, A_{5s}^{2f} and A_{5s}^{3f} remain at the value of "1" through their supports. Owing to this, they do not contribute to the firing strength in the respective rules, therefore x_8 and x_{10} can be removed from R_s^{1f}, x_4, x_5 and x_{10} from R_s^{2f}, x_{10} from R_s^{3f} and x_5 and x_8 from R_s^{4f}, and the fuzzy rules can be rewritten as follows:

R_s^{1f} : if x_4 is A_{1s}^{1f} $\wedge x_2$ is A_{2s}^{1f} $\wedge x_5$ is A_{3s}^{1f} $\wedge$ $\wedge x_1$ is A_{6s}^{1f} then y is $B_s^{1f}\left(f_s^{1f}(x), 20.62\right)$;

R_s^{2f} : if x_2 is A_{2s}^{2f} $\wedge$ x_8 is A_{4s}^{2f} $\wedge x_1$ is A_{6s}^{2f} then y is $B_s^{2f}\left(f_s^{2f}(x), 20.61\right)$;

R_s^{3f} : if x_4 is A_{1s}^{3f} $\wedge x_2$ is A_{2s}^{3f} $\wedge x_5$ is A_{3s}^{3f} $\wedge x_8$ is A_{4s}^{3f} $\wedge$ x_1 is A_{6s}^{3f} then y is $B_s^{3f}\left(f_s^{3f}(x), 20.61\right)$;

R_s^{4f} : if x_4 is A_{1s}^{4f} $\wedge x_2$ is A_{2s}^{4f} $\wedge$ x_{10} is A_{5s}^{4f} $\wedge x_1$ is A_{6s}^{4f} then y is $B_s^{4f}\left(f_s^{4f}(x), 20.62\right)$;

The value of the objective function J for the learning data is found to be 0.0010 while for the prediction data, J is 5.18×10^{-4}.

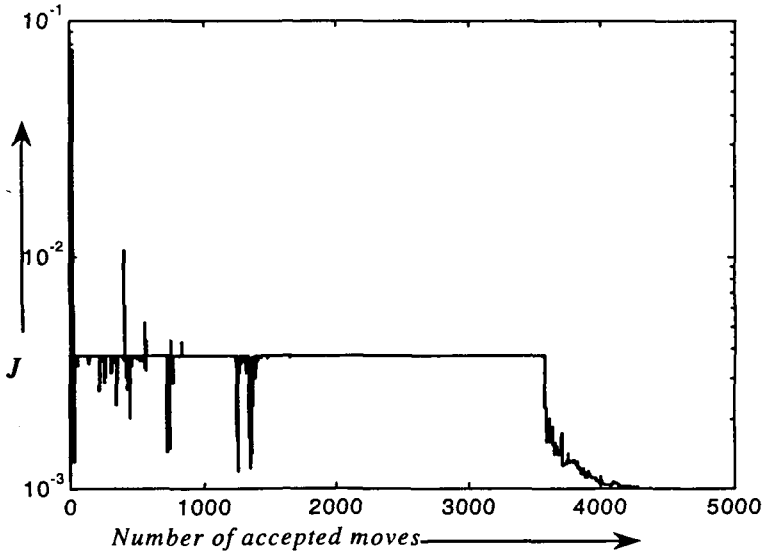


Fig. 5.18: Learning pattern of SA hybrid algorithm of *Example 2.4*.

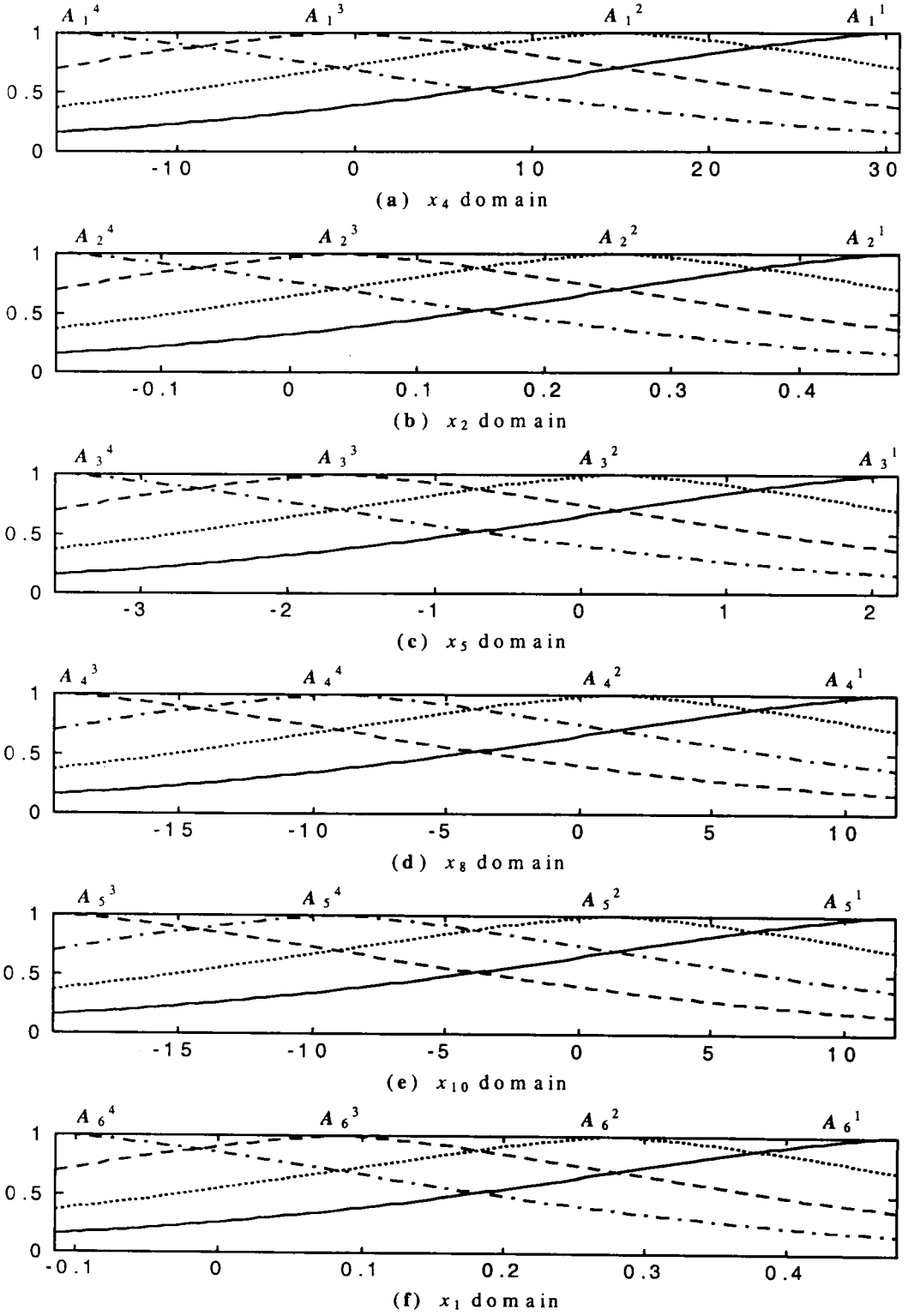


Fig. 5.19: Initial membership functions for class II GFM rules, obtained from fuzzy curve, of *Example 2.4.*

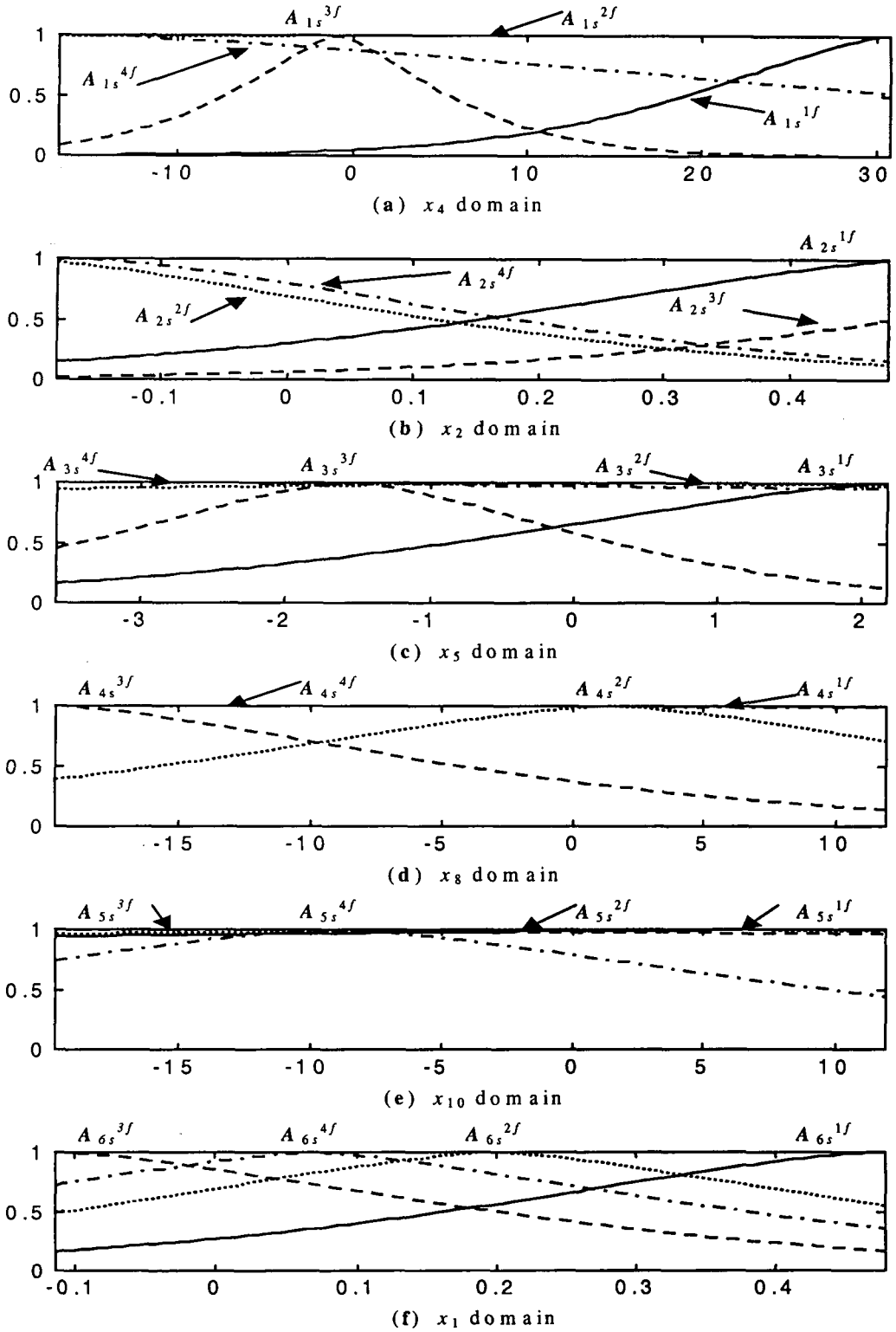


Fig. 5.20: Final membership functions for class II GFM rules, learned by SA hybrid algorithm, of *Example 2.4*.

Still the value of J does not reach the target value by the SA hybrid scheme, we have applied the GA hybrid algorithm with 100 populations. We have fed the initial parameters obtained by different methods of chapter 4 randomly to the GA hybrid algorithm. The maximum fitness in each generation is shown in Fig.5.20. Improvement in the solution is obtained over 68 generations. The value of J for the learning data is found to be 4.59×10^{-4} while for prediction data J is 6.98×10^{-4}.

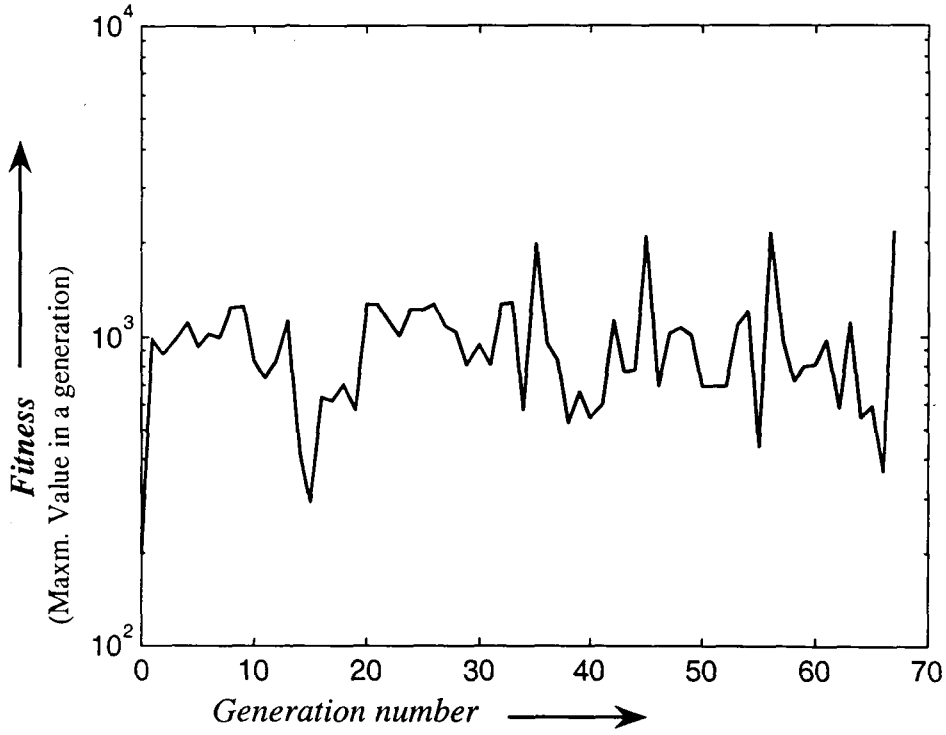


Fig. 5.21: Learning pattern of GA hybrid algorithm of *Example 2.4*.

Final fuzzy rules corresponding to the GFM learned by the hybrid GA scheme are listed below:

R_g^{1f} : if x_4 is A_{1g}^{1f} $\wedge x_2$ is A_{2g}^{1f} $\wedge x_5$ is A_{3g}^{1f} $\wedge x_8$ is A_{4g}^{1f} $\wedge x_{10}$ is A_{5g}^{1f} $\wedge x_1$ is A_{6g}^{1f} then y is $B_g^{1f}\left(f_g^{1f}(\boldsymbol{x}), 20.12\right)$;

R_g^{2f} : if x_4 is A_{1g}^{2f} $\wedge x_2$ is A_{2g}^{2f} $\wedge x_5$ is A_{3g}^{2f} $\wedge x_8$ is A_{4g}^{2f} $\wedge x_{10}$ is A_{5g}^{2f} $\wedge x_1$ is A_{6g}^{2f} then y is $B_g^{2f}\left(f_g^{2f}(\boldsymbol{x}), 20.12\right)$;

R_g^{3f} : if x_4 is A_{1g}^{3f} $\wedge x_2$ is A_{2g}^{3f} $\wedge x_5$ is A_{3g}^{3f} $\wedge x_8$ is A_{4g}^{3f} $\wedge x_{10}$ is A_{5g}^{3f} $\wedge x_1$ is A_{6g}^{3f} then y is $B_g^{3f}\left(f_g^{3f}(\boldsymbol{x}), 20.13\right)$;

R_g^{4f} : if x_4 is A_{1g}^{4f} $\wedge x_2$ is A_{2g}^{4f} $\wedge x_5$ is A_{3g}^{4f} $\wedge x_8$ is A_{4g}^{4f} $\wedge x_{10}$ is A_{5g}^{4f} $\wedge x_1$ is A_{6g}^{4f} then y is $B_g^{4f}\left(f_g^{4f}(\boldsymbol{x}), 20.13\right)$;

where,

$$f_g^{1f}(\boldsymbol{x}) = 346.50 + 10.04x_4 - 508.54x_2 - 113.02x_5 + 7.39x_8 + 23.20x_{10} - 769.84x_1$$

$$f_g^{2f}(\boldsymbol{x}) = -1.56 - 0.39x_4 - 49.79x_2 + 6.75x_5 - 1.24x_8 - 1.03x_{10} + 13.01x_1$$

$$f_g^{3f}(\boldsymbol{x}) = 7.24 + 0.03x_4 - 56.60x_2 - 9.99x_5 - 0.189x_8 + 1.08x_{10} + 3.79x_1$$

$$f_g^{4f}(\boldsymbol{x}) = -7.40 - 1.68x_4 + 95.81x_2 - 4.07x_5 + 0.36x_8 - 0.02x_{10} - 27.79x_1$$

The learned membership functions of the premise variables A_{1g}^{1f} to A_{1g}^{4f}, A_{2s}^{1f} to A_{2s}^{4f}, A_{3g}^{1f} to A_{3g}^{4f}, A_{4g}^{1f} to A_{4g}^{4f}, A_{5g}^{1f} to A_{5g}^{4f} and A_{6g}^{1f} to A_{6g}^{4f} are shown in Fig.5.22(a-f) respectively. The premise membership function A_{2g}^{3f} remains at the value of "0" over their supports, the rule R_g^{3f} is removed. The premise membership functions A_{1g}^{4f} and A_{3g}^{4f} remain at the value of "1" through their supports. Owing this they do not contribute to the firing strength in the respective rules, therefore x_4 and x_5 can be removed from R_g^{4f} and the fuzzy rules can be rewritten as follows:

R_g^{1f} : if x_4 is A_{1g}^{1f} $\wedge x_2$ is A_{2g}^{1f} $\wedge x_5$ is A_{3g}^{1f} $\wedge x_8$ is A_{4g}^{1f} $\wedge x_{10}$ is A_{5g}^{1f} $\wedge x_1$ is A_{6g}^{1f} then y is $B_g^{1f}\left(f_g^{1f}(\boldsymbol{x}), 20.12\right)$;

R_g^{2f} : if x_4 is A_{1g}^{2f} $\wedge x_2$ is A_{2g}^{2f} $\wedge x_5$ is A_{3g}^{2f} $\wedge x_8$ is A_{4g}^{2f} $\wedge x_{10}$ is A_{5g}^{2f} $\wedge x_1$ is A_{6g}^{2f} then y is $B_g^{2f}\left(f_g^{2f}(\boldsymbol{x}), 20.12\right)$;

~~R_g^{3f} : if x_4 is A_{1g}^{3f} $\wedge x_2$ is A_{2g}^{3f} $\wedge x_5$ is A_{3g}^{3f} $\wedge x_8$ is A_{4g}^{3f} $\wedge x_{10}$ is A_{5g}^{3f} $\wedge x_1$ is A_{6g}^{3f} then y is $B_g^{3f}\left(f_g^{3f}(\boldsymbol{x}), 20.13\right)$;~~

R_g^{4f} : if x_2 is A_{2g}^{4f} $\wedge$ x_8 is A_{4g}^{4f} $\wedge x_{10}$ is A_{5g}^{4f} $\wedge x_1$ is A_{6g}^{4f} then y is $B_g^{4f}\left(f_g^{4f}(\boldsymbol{x}), 20.13\right)$;

Figure 5.23 shows the actual output y, model output y^o and model error. From Fig.5.23(b) we can infer that the model predicts very well.

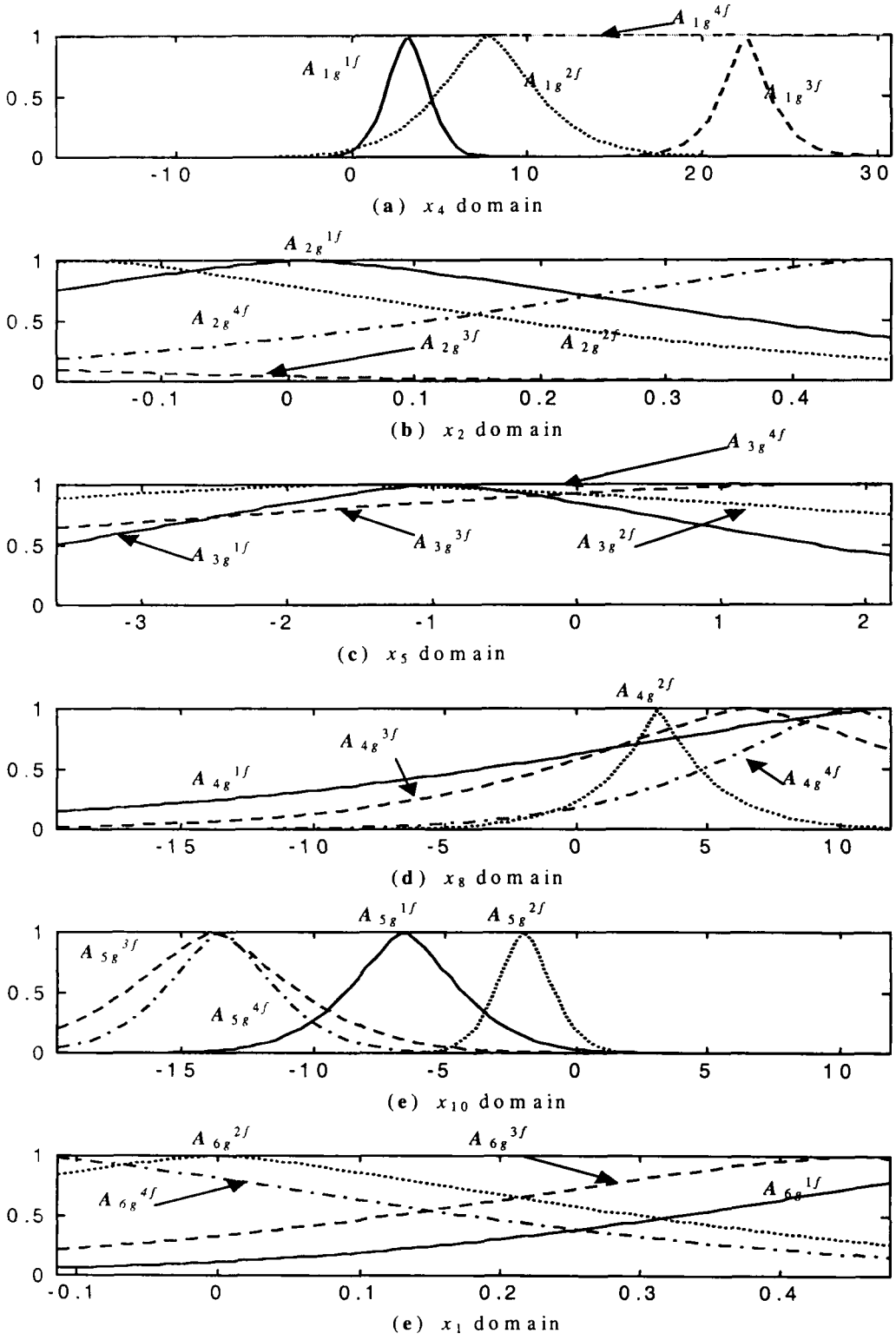


Fig. 5.22: Final membership functions for class II GFM rules, learned by GA hybrid algorithm, of *Example 2.4*.

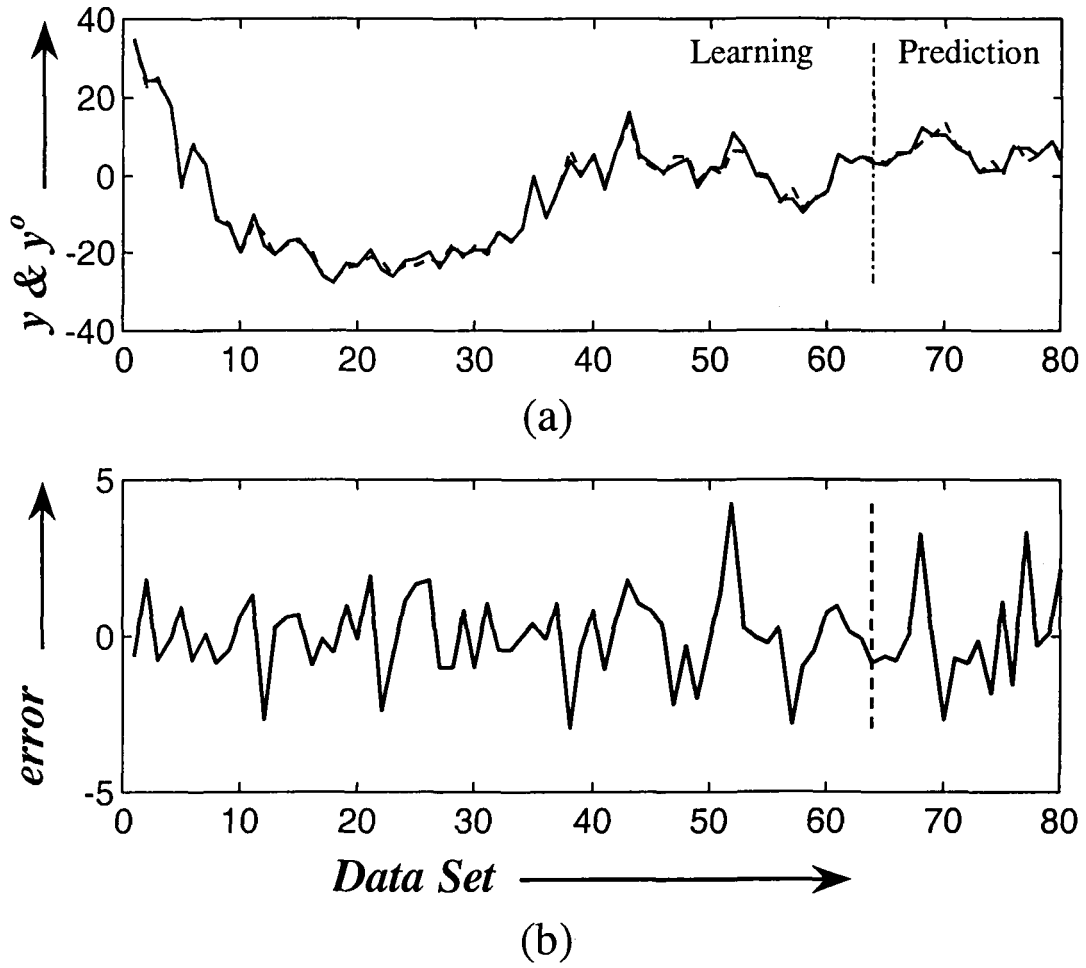


Fig.5.23: Plots of (a) y (————, solid line) and y^o (- - - - - -, dashed line) (b) *model error* for class II GFM of *Example 2.4* obtained from hybrid GA algorithm

5-7 Conclusions

This chapter presents the hybridization of learning algorithms for the estimation of consequent part of GFM rules in the LSE framework and premise part of GFM rules using GD learning. The local hybrid learning algorithm involving LSE and GD is then further hybridized separately with GA and SA to yield GA hybrid and SA hybrid respectively. The idea of hybridization is to improve the performance of the models in case the target performance is not achieved with the local learning algorithm. The initial

fuzzy rules of class II GFM are those obtained from fuzzy curve approach. It may be noted that if the membership function of a variable remains constant over the domain of the variable, then that variable is eliminated from the corresponding rules thus simplifying the model. The effectiveness of the GA hybrid on the enhancement of model performance with lesser number of rules, has been demonstrated on the *Example 2.4* of chapter 2

Chapter 6

MODELING OF FLUIDIZED CATALYTIC CRACKING UNIT

6-1 Introduction

A Fluidized Catalytic Cracking Unit (FCCU) is a typical complex system in process industry. FCCU consists of multiple sophisticated chemical and physical processes. The production goal of a FCCU is to produce the cracked light petro-oil products, e.g. gasoline, light diesel etc. through the catalytic chemical reaction of heavy oil. From a system point of view, FCCU can be considered as a high dimensional, highly nonlinear, interconnected, complex system.

Multivariable system identification of FCCU using linear regression technique has been reported in **[Donkelaar'98]**. Commercial models are also available as supplied by set point, DMC in USA etc. However, considering the unique process characteristics and especially its importance in the context of petroleum refinery industry in India, an endeavor has been made in this thesis to develop a neuro-fuzzy model of FCCU using hybrid learning techniques.

In the present work, we model the Reactor- Regenerator -Stripper - Fractionator (RRSF) section of FCCU using GRBF networks to estimate the parameters of fuzzy models. These models are based on the data collected from an operating refinery of FCCU with a capacity of 1.2 MMPTA.

The organization of this chapter is as follows: The process overview is briefly described in Section 6-2. The operating data of the process along with the variables involved in the operation of the process are marked in Section 6-3. A detailed modeling of all the variables comprising controlled and monitoring variables is presented in Section 6-4. Finally, conclusions of the whole modeling process are relegated to Section 6-5.

6-2 Process Overview

Fluid catalytic cracking (FCC) is considered as the primary conversion process in an integrated refinery. The FCC unit utilizes a microspherodial catalyst, which fluidizes when properly aerated. The main purpose of the unit is to convert high-boiling petroleum fractions called gas oil to high-value, high-octane gasoline and heating oil. Gas oil is the portion of crude oil that boils in the 650^0-1050^0+F (330^0-550^0C) range and contains a diversified mixture of paraffins, naphthenes, aromatics, and olefins.

Before proceeding, it is helpful to examine how a typical catalytic cracker fits into the refinery process. A petroleum refinery is composed of several processing units that are designed to convert raw crude oil into usable products such as gasoline, diesel, and jet fuel (Fig.6.1).

The crude unit is the first processing unit in the refining processes. Here, the raw crude is distilled into several intermediate products. The heavy portion of the crude oil

that cannot be distilled in the atmospheric tower is heated and sent to the vacuum tower. The tar from the vacuum tower is sent to the delayed coker, visbreaker, or other resid processing units for further processing.

The gas oil to a conventional catalytic cracker comes primarily via the atmospheric column, the vacuum tower, and the delayed coker unit. In addition, many refiners blend some atmospheric or vacuum residue with the cracker feed stocks to be processed in the FCC unit.

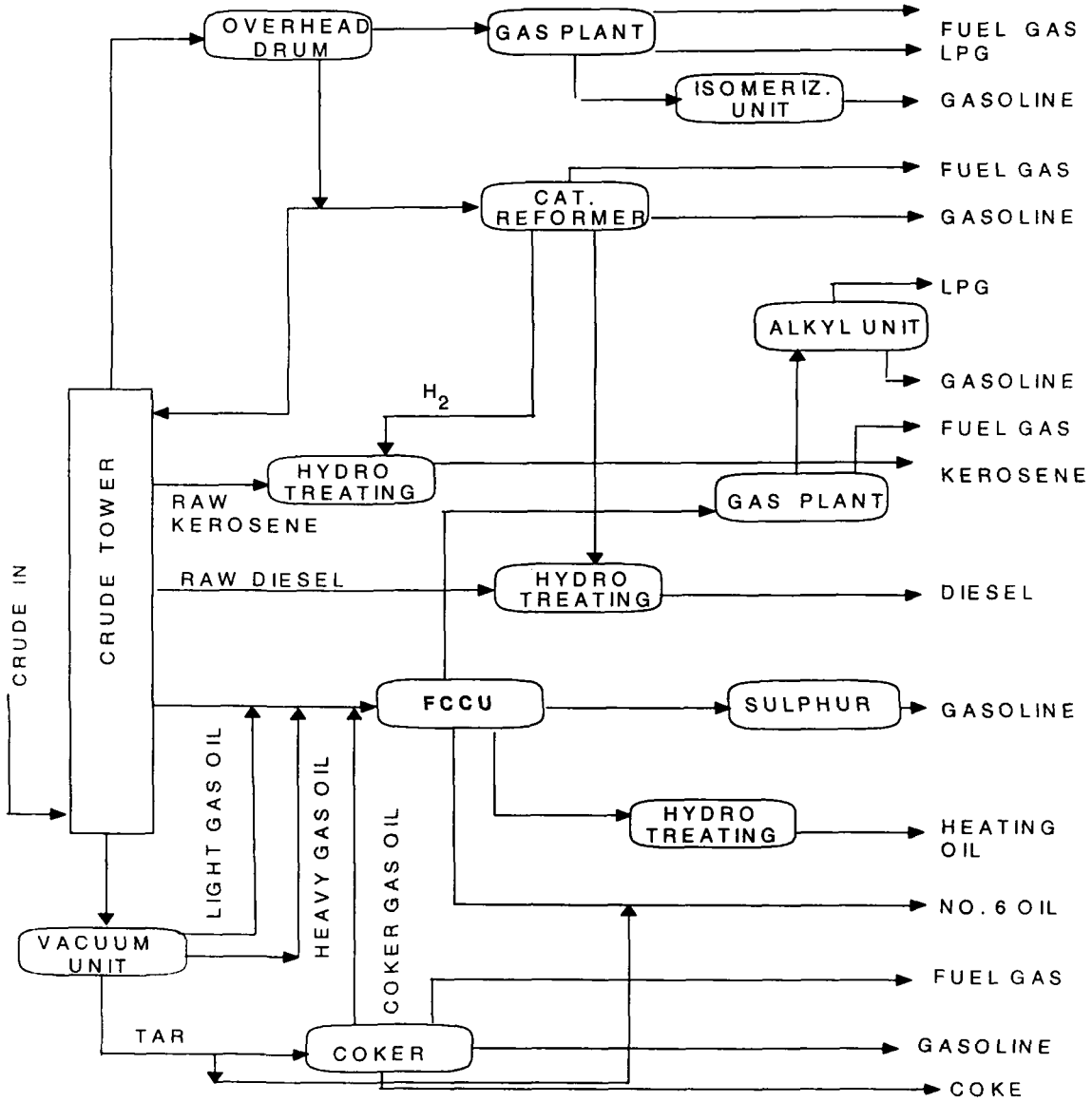


Fig. 6.1: A typical high conversion refinery.

The FCC process is very complex. To provide a clear understanding of operation of the unit, the process description can be broken down into four separate sections. These include the following:

- Feed Preheat
- Reactor
- Regenerator
- Main Fractionator

6-2.1 Feed Preheat

The refinery-produced gas oil and any supplemental feed stocks are generally combined and sent to a surge drum which provides a steady flow of feed to the FCC unit's charge pumps. The drum can also serve as device to separate any water or vapor that may be inherent in the feed stocks.

From the surge drum, the feed is normally heated to a temperature in the range, 550^0-700^0F (270^0-357^0C). The main fractionator bottoms pump-around and/or fired heaters are usually the sources of heat to preheat the feed. The feed preheater provides a tool to easily vary the catalyst-to-oil ratio. In units where air blower is the constraint, increasing preheat temperature allows increased throughput.

6-2.2 Reactor

The Reactor-Regenerator-Stripper (RRS), as shown in Fig.6.2, is the heart of the FCC process. In a modern catalytic cracker, virtually all the reactions occur in the riser over a short period of two to four seconds before the catalyst and the products are separated in the reactor.

From the preheater, the feed enters the riser near the base where it contacts with the incoming regenerated catalyst. The riser is essentially a vertical pipe usually having a 4 to 5 inch thick refractory lining for insulation and abrasion resistance. The ideal riser simulates a plug flow reactor, where the catalyst and the vapor travel along the length of the riser at the same velocity with minimum back mixing. Efficient contacting of the feed and the catalyst is critical for achieving the desired cracking reactions. Steam is commonly used to atomize the feed because feed atomization increases the availability of feed at the reactive acid sites on the catalyst. With the employment of a high-activity zeolite catalyst, virtually all of the cracking reactions take place in the riser in the time frame of less than two-second. In the Shell design there is also a vessel called *Stripper*, which strips off the catalyst from the coke with the help of steam.

(a) Catalyst Separation

After exiting the riser, the catalyst enters the reactor. In today's FCC operations, the reactor serves basically two functions: as a disengaging space for the separation of catalyst and vapor, and as the reactor cyclone's housing.

Most FCC units employ either a single cyclone or a two-stage cyclone to separate the remaining catalyst and return it to the stripper through the use of dipoles and flapper valves. The product vapors exit the cyclones and flow to the main fractionator column for recovery.

It is important to incorporate mechanical provisions to separate catalyst and vapors as soon as they enter the reactor, otherwise, the extended contact of vapors with the catalyst in the reactor will allow recracking of some of the desirable products.

Furthermore, the extended residence time also promotes thermal cracking of the desirable products.

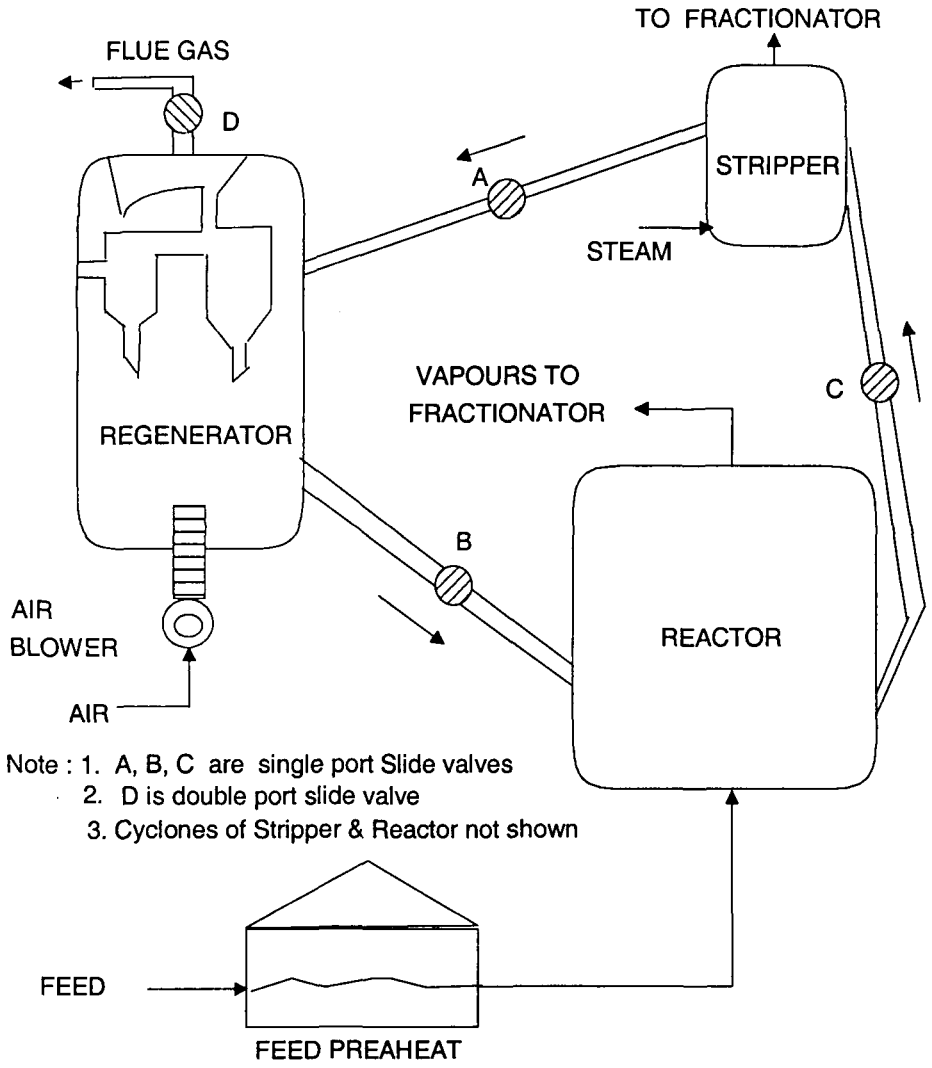


Fig.6. 2: A Typical Reactor-Regenerator-Stripper (RRS) Assembly

(b) Stripping Section

As the spent catalyst falls into the stripper, valuable hydrocarbons are adsorbed within the catalyst bed. Stripping steam, at a rate of 2-5 lbs. per 1,000 lbs. of circulating catalyst, is used to strip these hydrocarbons from the catalyst. Both baffled and unbaffled stripper designs are used in commercial contacting between the catalyst and steam.

It should be noted that not all the hydrocarbon vapors could be displaced from the catalyst pores in the stripper. A fraction of the vapors is carried with the spent catalyst into the regenerator. These vapors have higher hydrogen to carbon ratio than the coke in the catalyst. The flow of the spent catalyst to the regenerator is typically controlled by the use of a valve that slides back and forth. This slide valve is used to control the catalyst level in the stripper.

6-2.3 Regenerator

The regenerator has two main functions: It restores the catalyst activity and supplies heat to crack the feed. The spent catalyst entering the regenerator contains Coke between 0.8 and 2.5 percent by weight, depending on the quality of the feed stocks. The components of coke are carbon, hydrogen, and trace amounts of sulphur and nitrogen. Air is the source of oxygen for the combustion of coke and is supplied by a large air blower.

There are two regions in the regenerator: the dense phase and the dilute phase. At the velocities common in the regenerator, 2-4 ft/sec, the bulk of catalyst particles are located in the dense bed immediately above the air distributor. The dilute phase is the region above the dense phase up to the cyclone inlet, and has a substantially low catalyst concentration.

(a) Standpipe/Slide Valve

From the regenerator, the regenerated catalyst flows down a transfer line commonly referred to as a standpipe, which provides the necessary pressure head needed to circulate the catalyst around the unit. In some units, standpipes are extended into the regenerator, and the top section is often called a catalyst hopper. The function of the

hopper is to provide a sufficient time for the regenerated catalyst to be aerated before entering the standpipe.

The flow rate of the regenerated catalyst to the riser is commonly regulated through the use of either a slide or plug valve. Its main function is to supply enough catalyst to heat the feed and achieve the desired reactor temperature.

(b) Catalyst Separation

The hot flue gas leaving the regenerator plenum holds an appreciable amount of energy. A number of heat recovery schemes is used to recover this energy. In most units, the flue gas pressure is let down to atmospheric pressure across an orifice chamber. The orifice chamber is a vessel containing a series of perforated plates designed to maintain a given backpressure upstream of the regenerator pressure control valve.

A power recovery train employing a turbo expander usually consists of four parts: the expander, a motor/generator, an air blower, and a steam turbine. The steam turbine is used primarily for startup and often to supplement the expander for generation of electricity. From the expander, the flue gas goes through a steam generator to recover additional energy. Depending on local environment regulations, an electrostatic precipitator (ESP) or a wet gas scrubber may be placed downstream of the waste heat generator prior to the release of the flue gas to the atmosphere. In some units, an ESP is used to further remove catalyst fines in the range of 5-20 μ from the flue gas. In other units, a wet gas scrubber is employed to remove both catalyst fines and sulphur compounds from the flue gas stream.

(c) Partial versus Complete Combustion

There are two methods of regeneration practiced by FCC units: *partial combustion and complete combustion.* In the partial combustion mode, regulation of the flow of combustion air controls the regenerator temperature. In the full combustion, or high-temperature, regeneration, the design temperature of most of the regenerator cyclone supports is limited to around 1250^0F. The full combustion mode of regeneration utilizes excess oxygen for complete CO combustion and reduction of carbon in the regenerated catalyst is restricted to less than 0.10 wt%. Full combustion can be achieved either thermally or with the aid of a combustion promoter. Most catalytic crackers use a CO combustion promoter catalyst to encourage CO combustion in the dense phase.

(d) Catalyst Handling Facilities

Even with proper operation of the reactor and regenerator cyclones, catalyst particles smaller than 20 microns still escape from both of these vessels. The catalyst fines escaping the reactor collect at the fractionator bottom product storage tank. The electrostatic precipitator removes the recoverable catalyst fines exiting the regenerator.

The circulating catalyst in the FCC unit is called equilibrium catalyst, or simply E-cat. Periodically, quantities of equilibrium catalyst are withdrawn and stored in the E-cat hopper for future disposal. A refinery that processes the residue feed stocks in its catalytic cracker can make use of a good-quality E-cat from another refinery that processes the light sweet feed. The residue feed stocks contain large quantities of impurities such as metals. Therefore, the use of a good-quality E-cat in conjunction with the fresh catalyst can be cost-effective in maintaining low costs.

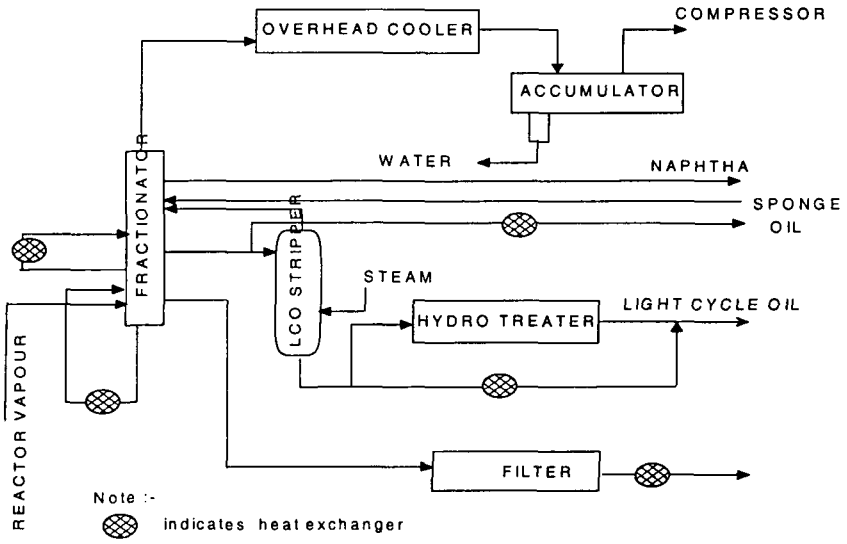


Fig. 6.3: A typical FCC main fractionator circuit.

6-2.4 Main Fractionator

The purpose of the main fractionator (shown in Fig.6.3) is to de-superheat and recover vapor. The hot-product vapors from the reactor flow into the main fractionator; these vapors enter the column near the base. The main function of the fractionator is to condense and separate the reaction products. The operation of the main column is similar to a crude tower but with two differences. First, the effluent vapors must be cooled before any fractionation begins. Second, large quantities of gases will go overhead with the unstabilized gasoline for further separation.

The main purpose of the bottom section of the main column is to provide a heat transfer zone. The recovered heat from the main column bottoms is used to preheat the fresh feed, generate steam, and serve as a heating medium for the gas plant re-boilers. The heaviest products from the main column are commonly called slurry or decant oil. Aside from decanted oil product, the main column is often designed to have three possible side cuts: Heavy Cycle Oil (HCO), Light Cycle Oil (LCO), and heavy naphtha.

6-3 Process Data

The main operating parameters of the FCC plant are delineated in the Table 6-1 given below:

Table 6-1: Major Operating Conditions

Parameters		Quantity
GENERAL SPECIFICATION		
Design feed rate	MT/Day	3,500
Feed rate	MT/Day	2,825
CFR		1.26
Reactor temp	°C	514.2
Regenerator temp	°C	666.1
Stripping steam	MT/Day	50
Cat Circulation rate	MT/Min	12.3
H_2 in coke	wt%	8.7
CO_2/CO ratio		Complete combustion
FEED PROPERTIES		
Type of feed composition		BH
Feed distillation range (IBP/5%/95%/FBP)		(-/330/470/-)
Feed metal wt%		3.9
Feed CCR wt%		0.4
YIELD PATTERNS in (wt%)		
Gas		4.0
LPG		13.9
Gasoline		39.3
LCO		27.2
HCO		4.9
CLO		5.9
Coke + Loss		4.6
Total		100
CATALYST :		
Catalyst used		
Additives		CO promoter Metal passivator Octane booster
Cat loss/make up rate (MT/Day) on fresh catalyst		0.3 / 0.7
CRC	wt%	0.1

The modeling problem, in fact, covers two major processes: - Catalytic Cracking which involves chemical reaction dynamics and fractionator which involves distillation dynamics.

The objective of a modern short contact time FCC is frequently to maximize the yields of high-octane gasoline. It does so in a highly flexible manner, operating within a framework of heat, pressure, and chemical balances **[Whittington'72].**

6-3.1 Effect of Operating Variables on Performance

(a). The Heat Balance as an Overall Frame of Reference

The adiabatic FCC heat balance is derived from the concept of macroscopic energy balance of classical chemical engineering, i.e.,

Accumulation of energy within the system	=	transfer of energy into the system through the system boundary	-	transfer of energy out of the system through the system boundary
	+	energy generation within the system	-	energy consumption within the system

To make a total energy balance, the first law of thermodynamics (conservation of energy) requires the last two terms to be zero, and there is no accumulation of energy for all steady-state operations (by definition). The above equation may then be rewritten as:

Transfer of total energy into the system through the system boundaries	-	transfer of total energy out of the system through the system boundaries	= 0

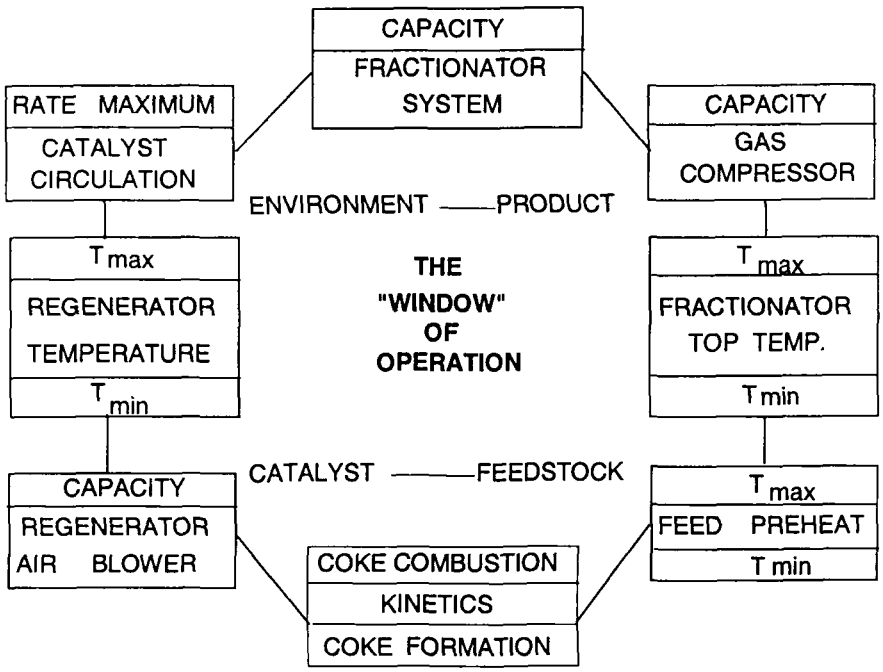


Fig.6.4: Engineering & Operation Constraints of FCCU.

Figure 6.4 shows some of the major engineering and operational constraints imposed on a fluid catalytic cracker. The rate of coke formation and combustion of coke with catalyst are not classified as operation variables, but are shown because of their critical coupling role in the reactor-regenerator dynamics.

Neglecting the very small kinetic energy terms, it can be simply stated that together the sum of the enthalpies of the reactants, minus the sum of the enthalpies of the products, minus the heat losses (by radiation and conduction) is equal to zero. The reactor outlet temperature and the flue gas temperature can be taken as a frame of reference for the reactor and regenerator energy balances, respectively. This simplifies the calculation and directly produces equations in which the heat of reaction terms are calculated at those respective reference temperatures. Accordingly, the simplified expressions for the heat balance are as follows:

For Reactor

$$-(CCR))\,(C_{pc})\,(\Delta T) + (FF)\,(CFR)\,(\Delta H_{CF}) + (SR)\,(\Delta H_S) + (FF)\,(\Delta H_{RX}) + (CKR)\,(\Delta H_{CS}) + L_{RX} = 0 \quad \ldots(6.1)$$

For Regenerator

$$-(CCR))\,(C_{pc})\,(\Delta T) + (CKR)\,(\Delta H_{CB}) + (AR)\,(C_{Pa})\,(T_{FG} - T_{AI}) - (CKR)\,(\Delta H_{CS}) + (CKR)\,(\Delta H_C) - COR\,(\Delta H_{CO}) = 0 \quad \ldots(6.2)$$

where

CCR	=	catalyst circulation rate, lb/hr
C_{pc}	=	catalyst heat capacity, Btu/lb/^{0}F
ΔT	=	temperature difference between regenerator dense bed and reactor dense bed/riser top, ^{0}F
CFR	=	combined feed ratio
ΔH_{CF}	=	enthalpy of combined feed at the reactor outlet temperature relative to the feed temperature, Btu/lb
SR	=	stripping steam rate, lb/hr
ΔH_S	=	enthalpy of stripping steam at reactor outlet temperature relative to steam inlet temperature, Btu/lb
FF	=	fresh feed rate, lb/hr
ΔH_{RX}	=	heat of reaction at reactor outlet temperature, Btu/lb
CKR	=	coke formation rate, lb/hr
ΔH_{CS}	=	heat of adsorption of coke at reactor outlet temperature, Btu/lb (exothermic, negative)
L_{RX}	=	heat losses from reactor by radiation and conduction, Btu/hr
ΔH_{CB}	=	coke heat of combustion at flue gas temperature, Btu/lb (exothermic, negative)
AR	=	Combustion air rate, lb/hr
Cp_a	=	air mean heat capacity, Btu/lb/^{0} F
TFG	=	regeneration flue gas temperature, ^{0}F
TAI	=	air inlet temperature, ^{0}F
L_{RG}	=	heat losses from regenerator by radiation and conduction, Btu/hr
ΔH_C	=	specific enthalpy of coke at flue gas temperature relative to reactor outlet temperature, Btu/lb
COR	=	CO rate in flue gas, lb/h
ΔH_{CO}	=	heat of combustion of CO at the gas temperature, Btu/lb (exothermic, negative)

If the eqns.(6.1) and (6.2) are solved for Catalytic Circulation Rate (CCR) and by equating these two equations, a relationship expressing coke formation rate (CKR) as a function of the independent operating variables will emerge **[Venuto'79].**

As schematically depicted in Fig.6.5, the reactor zone is physically and operationally coupled to the regenerator. The unit is "heat-balanced" in the sense that the heat for vaporizing and reacting the feed is derived by the combustion of the coke produced in the regenerator, and the released heat is transferred by the catalyst circulation to the reactor. An overall energy balance shows that the energy released by the combustion of coke becomes sensible to the latent heat of the reactor effluent and flue gas. Expressed another way, the coke yield in an adiabatic FCC is essentially that required to satisfy the heat load. Above and beyond feed preheat, commercial cracking units (excluding the distillation column) generally do not require heat exchange with external heat sources or sinks.

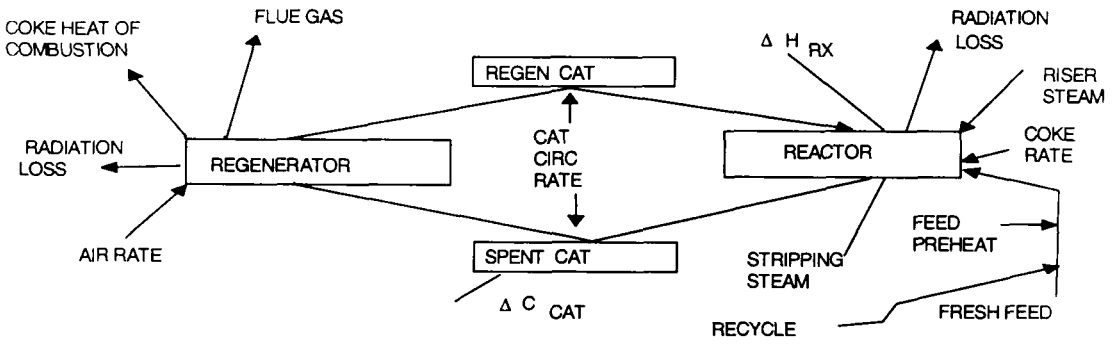


Fig.6.5: Heat balance in Reactor Regenerator Stripper.

The amount of coke burned may be calculated from the air rate and flue gas analysis, with the coke hydrogen content determined by oxygen disappearance. Use of

feed with the preheat, i.e., addition of thermal energy to the front end of the process, can be used to supply a portion of the heat needed by the reactor system and is particularly useful in utilizing the low coke-forming potentialities of zeolite catalysts. The chemical energy of the flue gas in the regenerator effluent is recovered at the back end via an external CO-boiler/power recovery system.

With the conventional FCC catalysts, about half of the carbon burned in the regenerator appears as flue gas CO, the ratio CO_2/CO typically being about 1:1. With hydro-treated feeds, the constraints of heat balance may occasionally influence the degree to which coke may be reduced (and hence the severity of hydro-treating) before requiring offsetting heat inputs from the feed preheat, etc.

b) The Key Reaction Variables

Some discussion of reaction variables can be found in literature **[Venuto'79; Lu'91]**. The major independent reaction variables in a commercial FCC, operating in the maximum gasoline mode are:

- Reactor temperature
- Combined feed rate (the sum of fresh feed and recycles rates) which controls the catalyst/oil residence time.
- Combined feed (feed preheat) temperature
- Reactor pressure
- Catalyst activity

These are directly regulated, usually with control devices. The dependent variables, on the other hand, are responsive to changes in the independent variables. These include:

- The catalyst circulation rate, CCR (and hence the Catalyst/Oil, {C/O} ratio)
- The regenerator temperature
- Conversion

The latter is subject to the influence by many independent variables. The air rate required to support the combustion of coke deposits may also be classified as a dependent variable. Of course, feedstock variations will profoundly affect all the variables.

In contrast, with most pilot plants, variables can be changed independently of heat. Here, reactor and regenerator temperatures may both be independently controlled, and greater flexibility for experimental work is allowed, particularly in that the C/O ratio. The CCR can be used as an independent operating variable to control the reaction severity. We now briefly describe some important variables.

*i) **Reactor Temperature:*** Catalytic cracking, at higher temperatures and short contact times, is generally associated with the higher gasoline octane numbers and the greater production of light olefins, especially C3 and C4 olefins - with their potential for alkylation feed. Cracked gasoline from high-temperature operations is a particularly useful blending component for lead-free gasoline. With short contact time riser cracking, the improvement in octane can be achieved without appreciable overcracking of gasoline.

A characteristic shift in product distribution with increasing temperature (at constant conversion) is observed in the riser FCC pilot plant while cracking 28.4° API gas oil. As the reactor temperature increases, dry gas and total C4's increase, but gasoline and coke decrease. Trends in the composition of dry gas, total C4's, and $C5^+$ gasoline with increasing temperature show that there is an increase in olefin yields of every fraction; and is particularly sharp in the C3 to C5 range.

*ii) **Combined Feed Rate:*** The importance of combined feed rate (the sum of fresh and recycle rates) for heat balance was already pointed out. Before the advent of zeolites, high recycle ratios were used to compensate for the poor selectivity. The first pass

conversions were low, with recycle essentially allowing longer catalyst/oil contact time for the unconverted material. This, of course, had the disadvantage of lowering the fresh feed capacity.

iii) Combined Feed (Feed Preheat) Temperature: The heat balance relationships emphasize the importance of a properly adjusted fresh feed temperature (See Fig.6.4) and combined feed ratio. Use of feed preheat (FPH) allows a reduction of C/O ratio, better selectivity to liquid products, and increased unit capacity. Combined feed temperatures may range up to 700 to 750^{0}C, with feed vaporization ranging up to 75% or higher in many fluid units (depending on the feed boiling range) like thermal cracking, may lead to furnace cooking and fouling.

iv) Catalyst/Oil Residence Time: Experience has shown that the most efficient way to use high activity zeolite catalysts is with short catalyst and oil contact time. Exposure of a few (1 to 4) seconds in the riser cracking is commonly all that is required to effect nearly complete conversion of the non-aromatic part of a feed, although this may vary with feedstock characteristics and yield objectives. At short contact times, selectivity for gasoline yield is maximized since the opportunity for secondary cracking of gasoline (over-cracking) is permitted less.

v) Pressure: Although it is, in fact, an independent variable, reactor pressure is often fixed by the capacity of unit and main column or gas compressor limitations. It is also to be noted that pressure has little effect on conversion or selectivity over the ranges commonly employed.

vi) Catalyst Activity: The type and activity of FCC catalyst are of great importance. Quality and activity must be maintained by an appropriate rate of addition of fresh

catalyst make-up, and are a direct reflection of the fraction of total inventory replaced per day.

vii) Catalyst Circulation Rate, Regenerator Temperature and Conversion: We may note here that in units functioning under controls, CCR and C/O ratio are not subject to direct control, and may be determined from the heat balance over either the reactor section or the regenerator section. Of central importance are the mutually dependent interactions among the feed preheat (FPH), CCR, and regenerator temperature (T_{RG}).

The converse of the above sequence of interactions also occurs, i.e., as FPH is decreased, ΔT (regenerator - reactor) decreases and coke-make increases; a similar pattern occurs if catalyst cooling or spray water addition to the regenerator is practiced rather than lowering FPH. In that case the system acts as a "flywheel" in which "coke rate" opposes the direction in which the regenerator temperature moves. The T_{RG} is thus stabilized under large fluctuations in feed stocks and operating conditions. In another sense the objective is to balance FPH with the coke-forming tendencies of the feedstock to achieve a high T_{RG}. Generally, higher FPH is required with better quality of feed stocks.

The lower coke rate, obtained with higher FPH, is also reinforced by the increased inefficiency of stripping. With the decline in CCR, not only is there less entrained hydrocarbon vapor entering the stripper, but also the catalyst residence time in the stripping zone is prolonged. Thus, more of the hydrocarbon vapor entrained in the interstices and void volumes of the catalyst particles is swept upward to join the reactor effluent and less hydrogen-rich material is carried from the reactor for combustion with (nonvolatile) coke -on- catalyst. The lower CCR may also be associated with less

attrition loss and lower quantities of CO_2 (entrained with the regenerated catalyst) that could ultimately cause scaling problems (as carbonate) in downstream gas plants.

Having considered the RRS section of FCCU let us now find out the main control variables for fractionator from the point of view of plant operations.

Fractionator: This is basically a distillation column. For a binary distillation column the number of variables are: -

Tray compositions (x_n and y_n)	=	$2N_T$
Tray liquid flows (L_n)	=	N_T
Tray liquid holdups (M_n)	=	N_T
Reflux drum composition (X_D)	=	1
Reflux drum flows (R and D)	=	2
Reflux drum holdup M_D	=	1
Reboiler compositions (x_B and y_B)	=	2
Reboiler flows (V and B)	=	2
Reboiler holdup (M_B)	=	1

The total number of variables is		4NT + 9

where

x_n = Liquid composition on n^{th} tray

y_n = Vapor composition on n^{th} tray

x = Relative volatility

N_T = No. of trays

whereas, the number of equations that can be written from the plant dynamics is $4N_T + 7$.

Therefore, the system is under-specified by the eqns.(6.1) and (6.2). From a control engineering viewpoint this means there are only two variables that can be controlled (can be fixed). The two variables that must somehow be specified are reflux flow R and vapor boilup (or heat input to the reboiler). They can be held constant (an

open-loop system) or they can be changed by a controller to hold any two variables constant.

However, the real life distillation column, as in this case, is multi-component and non-ideal. Typically the control objectives in a distillation operation are to maintain distillate composition and x_B at set point in the presence of disturbance. These disturbances may be characterized due to (1) loads, (2) changes in cooling and heating medium supply condition and (3) equipment fouling **[Deshpande'85].**

On the basis of the foregoing discussion, we may conclude that any distillation control system we design must be able to cope with the types of disturbances in the feed and in the supply conditions of the heat exchange devices just discussed.

From the above discussions, it is clear that the parameters from the reaction process and those from the fractionator must be chosen so as to make the model suitable for advanced control of the FCCU. The details of such parameters are given in Table 6-2. The controlled variables (1-10 in Fig.6.6) are subjected to step inputs while the rest are the monitoring variables. Figure 6.6 gives the flow diagram identifying these variables.

For ease of manipulation, the controlled and monitoring variables are tagged with "C" and "M" respectively. A number is suffixed to these tags. In Fig.6.6, the parameters are identified in terms of their serial numbers. as appearing in the Table 6-2.

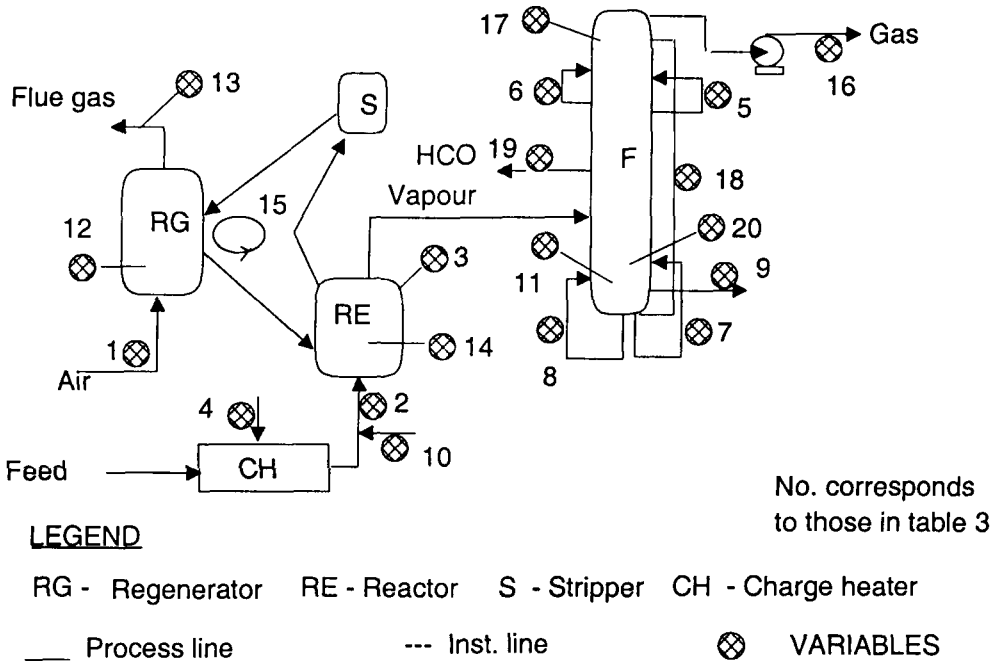


Fig. 6.6: Flow Diagram.

Table 6-2: List of Process Variables

S. No.	Service	Type of variable	Legend	S. No.	Service	Type of variable	Legend
1	Blower Air Flow to Regenerator (MT/Day)	Controlled input	C_1	11	Fractionator Bottom Level (%)	Monitoring output	M_1
2	Total Feed to Reactor (MT/Day)	Controlled input	C_2	12	Regenerator Bed Temperature (°C)	Monitoring output	M_2
3	Reactor Pressure (Kg/cm^2)	Controlled input	C_3	13	Flue Gas Temperature (°C)	Monitoring output	M_3
4	Charge Heater Gas fire (MT/Day)	Controlled input	C_4	14	Reactor Bed Temperature (°C)	Monitoring output	M_4
5	Main Fractionator Aux. Reflux (MT/Day)	Controlled input	C_5	15	Catalyst Circulation Rate (MT/Day)	Monitoring input	M_5
6	Main Reflux (MT/Day)	Controlled input	C_6	16	Compressor Amperage (amp)	Monitoring output	M_6
7	Column Bottoms Circulation (MT/Day)	Controlled input	C_7	17	Fractionator Top Temp. (°C)	Monitoring output	M_7
8	Column Bottoms Circulation (MT/Day)	Controlled input	C_8	18	Delta P across the Fractionator (Kg/cm^2)	Monitoring output	M_8
9	Clarified Oil Production (MT/Day)	Controlled output	C_9	19	Side Draw off temperature for HCO (°C)	Monitoring output	M_9
10	Cold Feed to Reactor (MT/Day)	Controlled input	C_{10}	20	Bottom Vapor Temperature (°C)	Monitoring output	M_{10}

The data were collected from DCS (Distributed Control System) already operating in the plant. For each variable 12071 data were collected at an interval of one minute. The designed dynamic response time of the plant is about 2 hours, so this rate of collection of data is though suitable for developing the model, but the data handled would be enormous. Hence, the data after each 10 minutes interval were considered for modeling. This way, 1200 data points would be available representing a time span of 12000 minutes. In order to meet the dynamic response time of 2 hours, we consider upto five delays with the current samples for each variable. These five delays give rise to six samples at instant t, t-1, t-2, t-3, t-4, t-5. Since there are 20 variables it results in 120 input candidates for the evaluation of their significance.

Step inputs were given to each controlled variable in turn. After waiting for the plant to stabilize the next controlled variable was subjected to step change. The process was continued till the entire controlled variables were subjected to step changes. Meanwhile, data for both controlled and monitoring variables were collected. These controlled variables were kept in a reasonably constant value in between the step changes whereas monitoring variables were allowed to change as long as these were within the safety limits. Moreover, it is reasonable to consider that the RRS section variables, both controlled and monitoring variables, are comparatively less affected by the process parameters of Fractionator section. However, an upset in the pressure balance in the fractionator would result in considerable change in the parameters of the RRS section also. This is because it directly affects the withdrawal of the vapors formed as a result of the reaction. This is indeed manifested in almost all the process variables during the time when there is a sudden failure of the compressor resulting in variable M_6 (Compressor amperage) forced to zero at 2767 th instant or 276th data point.

6-4 Simulation Results

We consider 120 input candidates for building the class II GFM for each variable. The input selection is based on the criteria derived from the fuzzy curve approach of chapter 2, and the selected inputs corresponding to the model for each variable are listed in the fourth column of Table 6-3. The number of rules for each model, obtained from a heuristic based on the fuzzy curve, is listed in the sixth column of Table 6-3 corresponding to the minimum value of the cluster validity function given in the fifth column of the table. The Modified Mountain clustering is used to form the clusters, which yield the initial fuzzy rules for the model. The model is then used for the initialization of network for supervised learning. A local algorithm composed of Gradient Descent and LSE trains the network. The parameters of the trained network are used in expressing the functionally equivalent fuzzy model. We have not used the hybrid algorithms for the simple reason that the performance of the most of the models happens to be satisfactory.

A detailed explanation of rules (GFM) for each variable is given. The modeling error is depicted to show the performance of the GFM. Also, the premise variables are analyzed whether they can be removed from the rule or not. As mentioned in chapter 5, the variable is removed if its response is flat and near to '1'. The results of this chapter are the culmination of efforts done in the previous chapters. It gives an idea of modeling of a real plant. Their interactive behavior is presented in the form of rules that are influenced by other variables.

The network is trained for the 80 percent of the data (i.e., 960), and tested for the rest of the data. The performance of the finely tuned model, obtained by the local algorithm, is listed in the last column of Table 6-3.

Table 6-3: List of Selected Input to FCCU variables Fuzzy Model and their Performance

S. No.	Variable	Associated unit	Input(s) to the Fuzzy Model	*S*	No. of Rules	*J* (learning/prediction)
1	$C_1(t)$	RRS	$C_1(t\text{-}1)$	1.65	4	$8.87\text{x}10^{-4}$/ $5.26\text{x}10^{-4}$
2	$C_2(t)$	RRS	$C_2(t\text{-}1)$, $M_8(t\text{-}1)$, $M_8(t)$, $M_8(t\text{-}3)$, $M_8(t\text{-}4)$	0.52	3	$2.51\text{x}10^{-4}$/ $2.22\text{x}10^{-4}$
3	$C_3(t)$	RRS	$M_7(t)$, $C_3(t\text{-}1)$, $C_3(t\text{-}2)$, $C_3(t\text{-}5)$	7.70	4	$8.12\text{x}10^{-4}$/ 0.0326
4	$C_4(t)$	RRS	$C_4(t\text{-}1)$, $C_4(t\text{-}2)$, $C_4(t\text{-}3)$, $M_6(t\text{-}3)$	0.33	3	$12.00\text{x}10^{-4}$/ $10.00\text{x}10^{-4}$
5	$C_5(t)$	Fract.	$M_{10}(t\text{-}5)$, $C_5(t\text{-}1)$, $C_5(t\text{-}2)$, $C_8(t)$	0.34	3	$8.97\text{x}10^{-4}$/ $4.77\text{x}10^{-4}$
6	$C_6(t)$	Fract.	$C_6(t\text{-}1)$	0.51	2	$25.00\text{x}10^{-4}$/ $17.00\text{x}10^{-4}$
7	$C_7(t)$	Fract.	$C_7(t\text{-}1)$, $M_5(t\text{-}3)$, $M_5(t\text{-}2)$	0.45	3	$5.01\text{x}10^{-4}$/ $1.54\text{x}10^{-4}$
8	$C_8(t)$	Fract.	$C_8(t\text{-}1)$, $C_5(t)$, $M_6(t\text{-}5)$, $M_6(t\text{-}3)$, $M_9(t\text{-}5)$,	0.45	3	$7.48\text{x}10^{-4}$/ $3.02\text{x}10^{-4}$
9	$C_9(t)$	Fract.	$C_9(t\text{-}1)$, $C_9(t\text{-}2)$, $M_1(t\text{-}2)$,	0.17	3	$1.79\text{x}10^{-2}$/ $1.32\text{x}10^{-2}$
10	$C_{10}(t)$	RRS	$C_{10}(t\text{-}1)$, $M_6(t)$, $M_6(t\text{-}1)$.	0.32	4	$4.84\text{x}10^{-4}$/ $4.79\text{x}10^{-4}$
11	$M_1(t)$	Fract.	$M_1(t\text{-}1)$, $M_1(t\text{-}2)$, $M_6(t\text{-}2)$, $M_6(t)$	0.34	3	$30.00\text{x}10^{-4}$/ $23.00\text{x}10^{-4}$
12	$M_2(t)$	RRS	$M_3(t\text{-}4)$, $M_2(t\text{-}5)$, $M_2(t\text{-}3)$, $M_2(t\text{-}2)$, $M_2(t\text{-}1)$	0.45	3	$9.49\text{x}10^{-5}$/ $3.52\text{x}10^{-5}$
13	$M_3(t)$	RRS	$M_3(t\text{-}1)$	1.81	3	$9.75\text{x}10^{-4}$/ $8.25\text{x}10^{-6}$
14	$M_4(t)$	RRS	$M_{10}(t)$, $M_4(t\text{-}1)$, $M_6(t)$, $M_4(t\text{-}2)$, $M_6(t\text{-}2)$, $M_8(t)$, $M_8(t\text{-}1)$, $M_8(t\text{-}4)$	0.26	3	$14.00\text{x}10^{-4}$/ $14.00\text{x}10^{-4}$
15	$M_5(t)$	RRS	$M_{10}(t)$, $M_9(t)$, $M_5(t\text{-}1)$, $M_2(t)$, $M_3(t\text{-}4)$	0.50	4	$14.00\text{x}10^{-4}$/ $7.52\text{x}10^{-4}$
16	$M_6(t)$	Fract.	$M_{10}(t)$, $M_6(t\text{-}1)$, $M_6(t\text{-}2)$	0.34	3	$14.00\text{x}10^{-4}$/ $31.00\text{x}10^{-4}$
17	$M_7(t)$	Fract	$M_{10}(t\text{-}2)$, $M_9(t\text{-}2)$, $M_6(t\text{-}2)$, $M_9(t)$, ~~$M_9(t\text{-}4)$~~, $M_9(t\text{-}3)$, $M_7(t\text{-}1)$	0.32	3	$7.40\text{x}10^{-4}$/ $2.72\text{x}10^{-4}$
18	$M_8(t)$	Fract.	$M_{10}(t)$, $M_{10}(t\text{-}2)$, $M_9(t)$, $M_8(t\text{-}1)$, $M_6(t)$, $M_9(t\text{-}2)$, $M_5(t\text{-}2)$	0.36	~~5~~ 4	$25.00\text{x}10^{-4}$/ $98.00\text{x}10^{-4}$
19	$M_9(t)$	Fract.	$M_{10}(t)$, $M_9(t\text{-}4)$, $M_9(t\text{-}3)$, $M_{10}(t\text{-}3)$, $M_9(t\text{-}2)$, $M_9(t\text{-}1)$	0.28	3	$12.00\text{x}10^{-4}$/ $6.49\text{x}10^{-4}$
20	$M_{10}(t)$	Fract.	$M_9(t)$, $M_5(t)$, $M_8(t)$, $M_6(t)$, $M_{10}(t\text{-}1)$, $M_{10}(t\text{-}3)$, $M_8(t\text{-}1)$	0.38	3	$8.52\text{x}10^{-4}$/ $5.17\text{x}10^{-4}$

In the identification stage, we do not make any distinction between controlled variable and monitoring variable. The identification of the structure and learning of parameters of all variables are given in the following:

Blower air flow to regenerator, $C_1(t)$

This variable is associated with RSS unit and it depends on the $C_1(t\text{-}1)$. Since the model input is only $C_1(t\text{-}1)$, the variable $C_1(t)$ exhibits the first order dynamics. The fuzzy rules are given as follows:

$$R^{1f}: \text{if } C_1(t-1) \text{ is } A_1^{1f} \quad \text{then } C_1(t) \text{ is } B^{1f}\left(f^{1f}(x), 19.47\right);$$

$$R^{2f}: \text{if } C_1(t-1) \text{ is } A_1^{2f} \quad \text{then } C_1(t) \text{ is } B^{2f}\left(f^{2f}(x), 19.47\right);$$

$$R^{3f}: \text{if } C_1(t-1) \text{ is } A_1^{3f} \quad \text{then } C_1(t) \text{ is } B^{3f}\left(f^{3f}(x), 19.47\right);$$

$$R^{4f}: \text{if } C_1(t-1) \text{ is } A_1^{4f} \quad \text{then } C_1(t) \text{ is } B^{4f}\left(f^{4f}(x), 19.47\right);$$

where,

$$f^{1f}(x) = 1292.00 - 0.91C_1(t-1)$$

$$f^{2f}(x) = 28547.00 - 11.56C_1(t-1)$$

$$f^{3f}(x) = 5863.00 - 2.20C_1(t-1)$$

$$f^{4f}(x) = -576.00 + 1.06C_1(t-1)$$

The premise variable membership functions A_1^{1f} to A_1^{4f} are shown in Fig.6.7

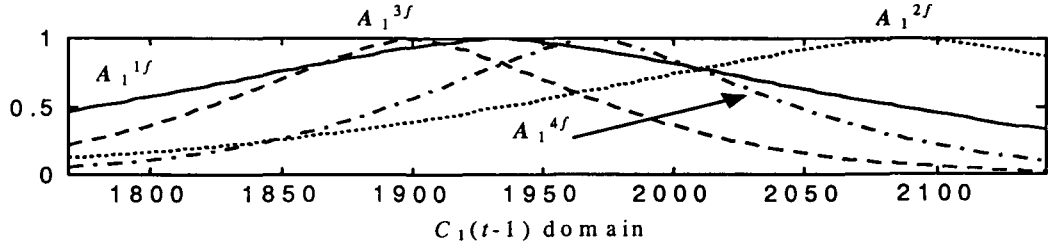


Fig. 6.7: Premise membership functions for class II GFM rules, of $C_1(t)$ variable.

The output of the model and the actual values are plotted in Fig.6.8(a). Figure 6.8(b) shows the model error in per unit.

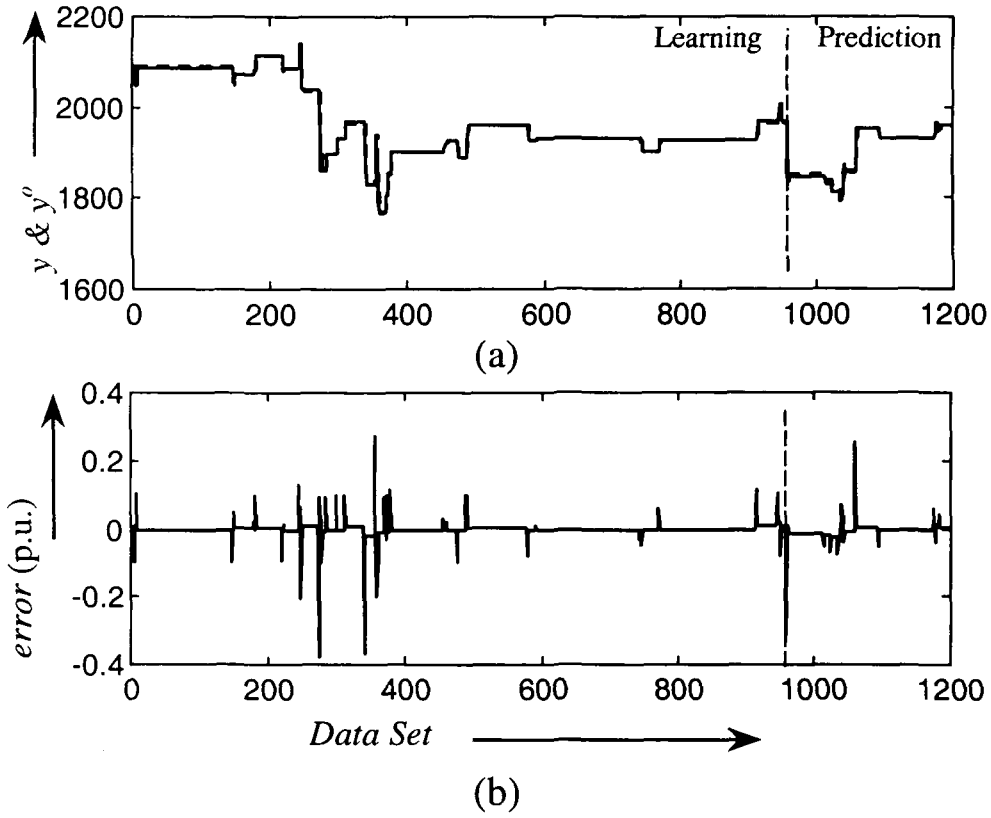


Fig.6.8: Plots of (a) y (————, solid line) and y^o (- - - - - -,dashed line) (b) *model error* for class II GFM of $C_1(t)$.

Total feed to Reactor, $C_2(t)$

This variable is associated with RRS unit and it depends on $C_2(t\text{-}1)$, $M_8(t\text{-}1)$, $M_8(t)$, $M_8(t\text{-}3)$ and $M_8(t\text{-}4)$ where . The variable M_8 is called the Delta P across the Fractionator. The variable $C_2(t)$ exhibits the first order dynamics, and the effect of instantaneous and delayed Delta P across the Fractionator. The fuzzy rules are given as follows:

$$R^{1f}: \text{if } C_2(t-1) \text{ is } A_1^{1f} \wedge M_2(t-1) \text{ is } A_2^{1f} \wedge M_2(t) \text{ is } A_3^{1f} \wedge M_2(t-3) \text{ is } A_4^{1f} \wedge M_2(t-4) \text{ is } A_5^{1f} \quad \text{then } C_2(t) \text{ is } B^{1f}\left(f^{1f}(x), 629.64\right);$$

R^{2f} :if $C_2(t-1)$ is $A_1^{2f} \wedge M_2(t-1)$ is $A_2^{2f} \wedge M_2(t)$ is $A_3^{2f} \wedge M_2(t-3)$ is $A_4^{2f} \wedge M_2(t-4)$ is A_5^{2f} then $C_2(t)$ is $B^{2f}\left(f^{2f}(x), 629.64\right)$;

R^{3f} :if $C_2(t-1)$ is $A_1^{3f} \wedge M_2(t-1)$ is $A_2^{3f} \wedge M_2(t)$ is $A_3^{3f} \wedge M_2(t-3)$ is $A_4^{3f} \wedge M_2(t-4)$ is A_5^{3f} then $C_2(t)$ is $B^{3f}\left(f^{3f}(x), 629.64\right)$;

where,

$$f^{1f}(x) = 1521.50 + 0.98C_2(t-1) + 20800.00M_8(t-1) - 31663.00M_8(t) - 6025.30M_8(t-3) - 2434.10M_8(t-4)$$

$$f^{2f}(x) = 1078.10 + 0.80C_2(t-1) - 4481.00M_8(t-1) + 3380.00M_8(t) - 925.00M_8(t-3) - 1366.90M_8(t-4)$$

$$f^{3f}(x) = -169.20 + 0.65C_2(t-1) - 23271.00M_8(t-1) + 29683.00M_8(t) + 4232.40M_8(t-3) + 468.00M_8(t-4)$$

The learned premise variable membership functions A_1^{1f} to A_5^{3f} are shown in Fig.6.9

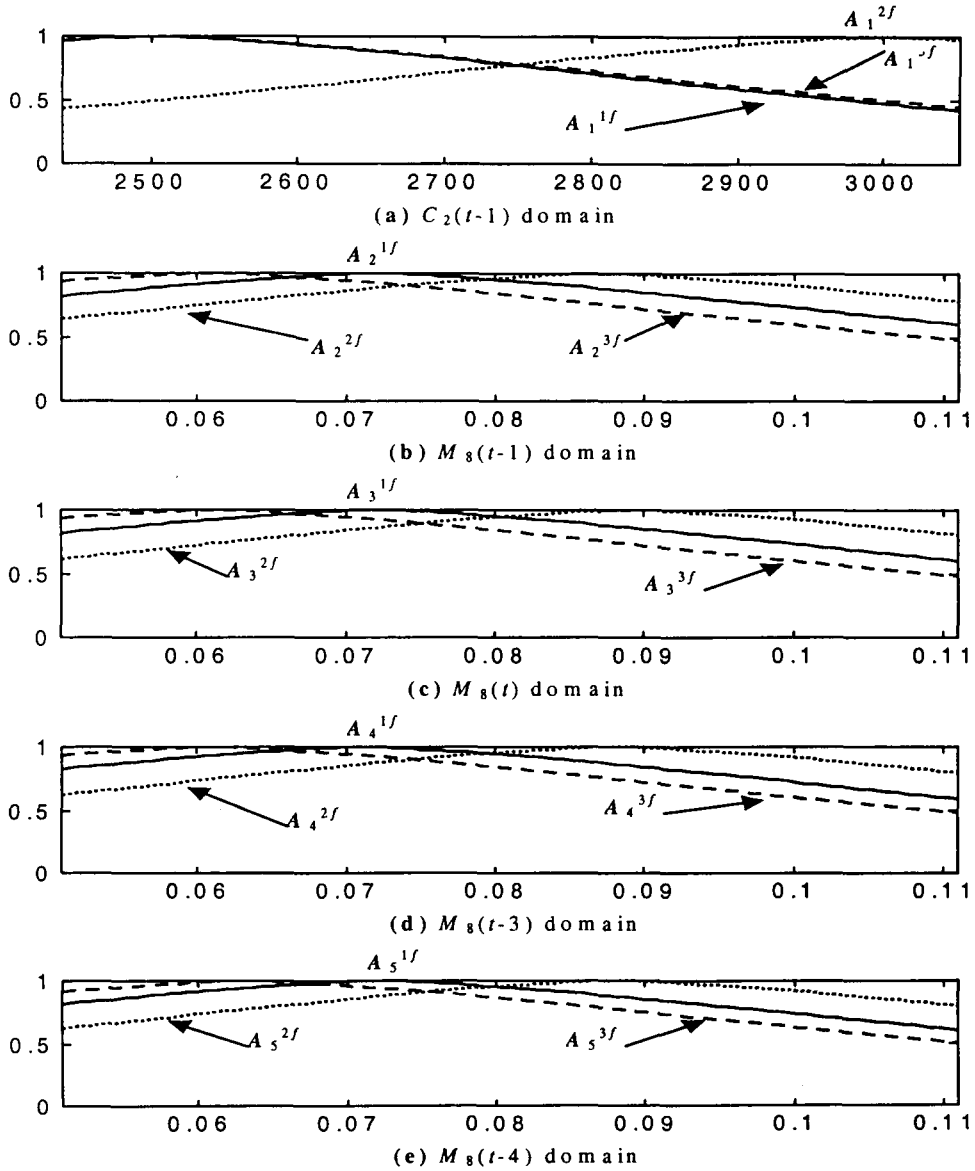


Fig. 6.9: Premise membership functions for class II GFM rules, of $C_2(t)$ variable.

The output of the model and the actual values are plotted in Fig.6.10(a). Figure 6.10(b) shows the model error in per unit.

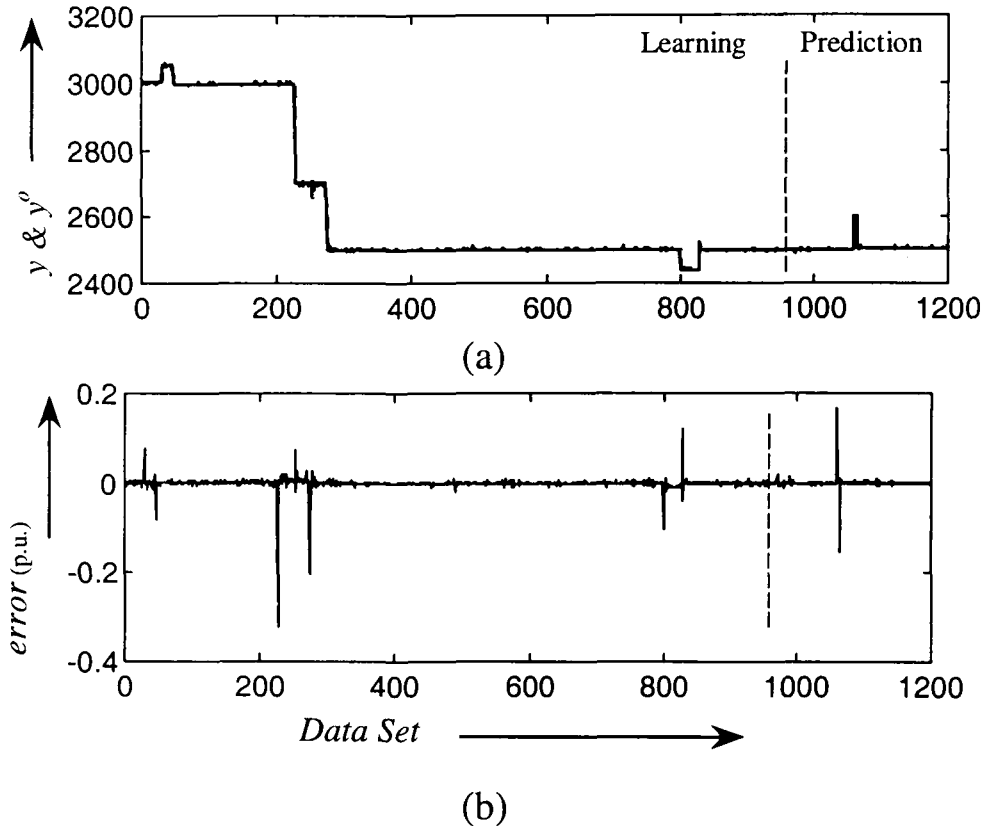


Fig.6.10: Plots of (a) y (_____, solid line) and y^o (- - - - - -,dashed line) (b) *model error* for class II GFM of $C_2(t)$.

Reactor Pressure, $C_3(t)$

This variable is associated with RRS unit and it depends on $M_7(t)$, $C_3(t\text{-}1)$, $C_3(t\text{-}2)$, and $C_3(t\text{-}5)$. The variable M_7 denotes the Fractionator Top Temperature. The variable $C_3(t)$ has fifth order dynamics and instantaneous effect of Fractionator Top Temperature. The fuzzy rules are given as follows:

R^{1f} :if $M_7(t-1)$ is $A_1^{1f} \wedge C_3(t-1)$ is $A_2^{1f} \wedge C_3(t-2)$ is $A_3^{1f} \wedge C_3(t-5)$ is A_4^{1f} then $C_3(t)$ is $B^{1f}\left(f^{1f}(x), 0.98\right)$;

R^{2f} :if $M_7(t-1)$ is $A_1^{2f} \wedge C_3(t-1)$ is $A_2^{2f} \wedge C_3(t-2)$ is $A_3^{2f} \wedge C_3(t-5)$ is A_4^{2f} then $C_3(t)$ is $B^{2f}\left(f^{2f}(x), 0.87\right)$;

R^{3f} :if $M_7(t-1)$ is $A_1^{3f} \wedge C_3(t-1)$ is $A_2^{3f} \wedge C_3(t-2)$ is $A_3^{3f} \wedge C_3(t-5)$ is A_4^{3f} then $C_3(t)$ is $B^{3f}\left(f^{3f}(x), 0.68\right)$;

R^{4f} :if $M_7(t-1)$ is $A_1^{4f} \wedge C_3(t-1)$ is $A_2^{4f} \wedge C_3(t-2)$ is $A_3^{4f} \wedge C_3(t-5)$ is A_4^{4f} then $C_3(t)$ is $B^{4f}\left(f^{4f}(x), 0.89\right)$;

where,

$$f^{1f}(x) = 3.52 + 0.01M_7(t) - 0.36C_3(t-1) + 8.49C_3(t-2) - 9.59C_3(t-5)$$

$$f^{2f}(x) = -3.01 - 0.01M_7(t) + 2.33C_3(t-1) - 0.29C_3(t-2) + 0.71C_3(t-5)$$

$$f^{3f}(x) = 0.84 + 0.93C_3(t-1) - 2.30C_3(t-2) + 2.09C_3(t-5)$$

$$f^{4f}(x) = -15.35 - 0.02M_7(t) + 0.03C_3(t-1) - 21.07C_3(t-2) + 30.64C_3(t-5)$$

The premise variable membership functions A_1^{1f} to A_4^{4f} are shown in Fig.6.11

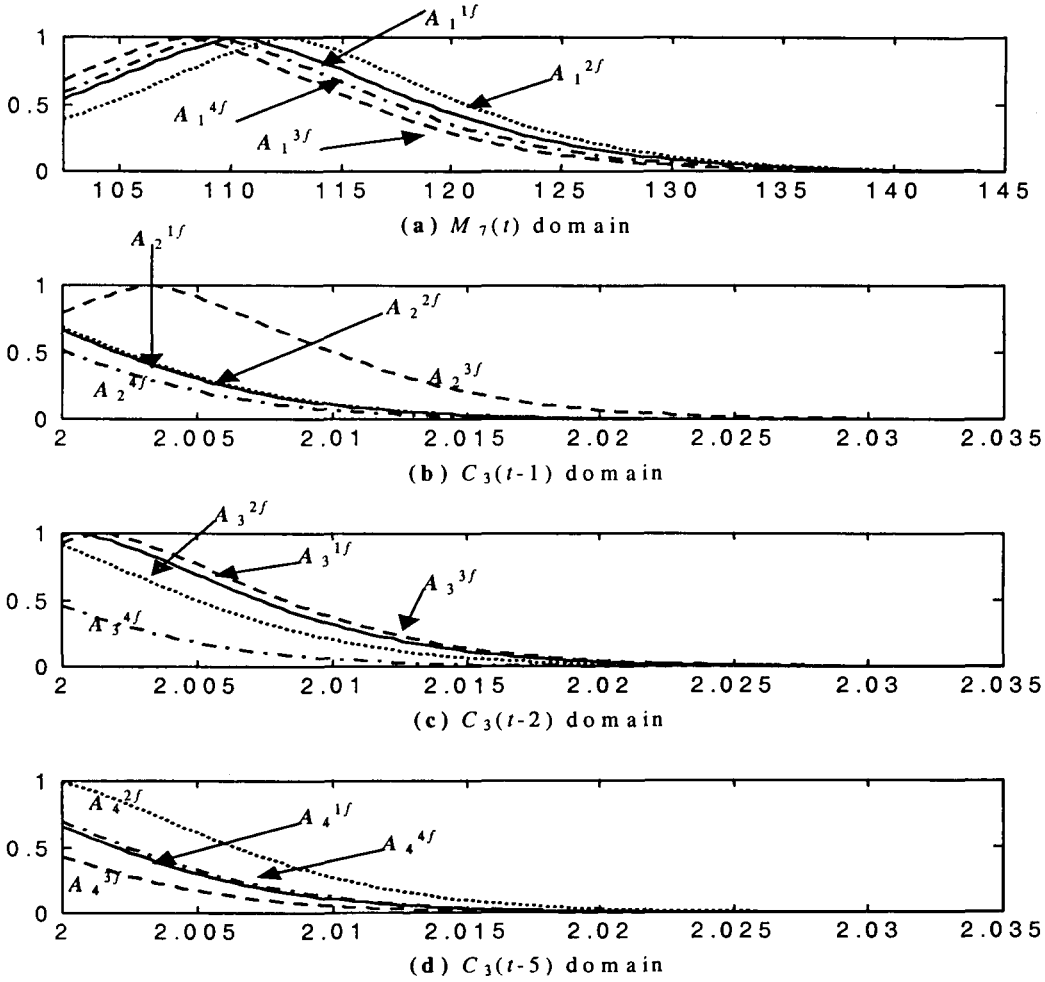


Fig. 6.11: Premise membership functions for class II GFM rules, of $C_3(t)$ variable

The output of the model and the actual values are plotted in Fig.6.12(a). Figure 6.12(b) shows the model error in per unit.

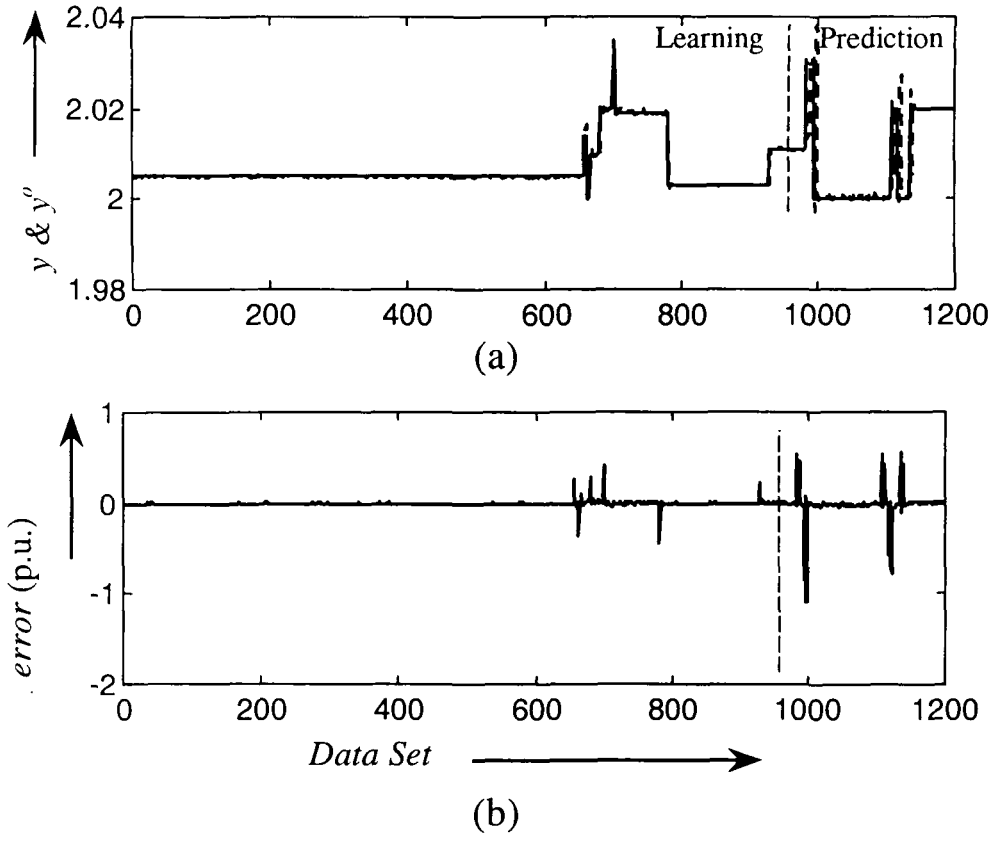


Fig.6.12: Plots of (a) y (————, solid line) and y^o (- - - - - -,dashed line) (b) *model error* for class II GFM of $C_3(t)$.

Charge Heater Gas Fire, $C_4(t)$

This variable is associated with RSS unit and it depends on $C_4(t\text{-}1)$, $C_4(t\text{-}2)$, $C_4(t\text{-}3)$, and $M_6(t\text{-}3)$. The variable $C_4(t)$ displays the third order dynamics with a 30 minutes delayed effect of Compressor Amperage, $M_6(t\text{-}3)$. The fuzzy rules are given as follows:

R^{1f} :if $C_4(t-1)$ is $A_1^{1f} \wedge C_4(t-2)$ is $A_2^{1f} \wedge C_4(t-3)$ is $A_3^{1f} \wedge M_6(t-3)$ is A_4^{1f} then $C_4(t)$ is $B^{1f}\left(f^{1f}(x), 3.23\right)$;

R^{2f} :if $C_4(t-1)$ is $A_1^{2f} \wedge C_4(t-2)$ is $A_2^{2f} \wedge C_4(t-3)$ is $A_3^{2f} \wedge M_6(t-3)$ is A_4^{2f} then $C_4(t)$ is $B^{2f}\left(f^{2f}(x), 3.29\right)$;

R^{3f} :if $C_4(t-1)$ is $A_1^{3f} \wedge C_4(t-2)$ is $A_2^{3f} \wedge C_4(t-3)$ is $A_3^{3f} \wedge M_6(t-3)$ is A_4^{3f} then $C_4(t)$ is $B^{3f}\left(f^{3f}(x), 3.28\right)$;

where,

$$f^{1f}(x) = -0.33 + 0.77C_4(t-1) + 0.55C_4(t-2) - 0.34C_4(t-3) + 0.002M_6(t-3)$$

$$f^{2f}(x) = 0.25 + 1.84C_4(t-1) - 1.42C_4(t-2) + 0.50C_4(t-3) + 0.002M_6(t-3)$$

$$f^{3f}(x) = -0.16 + 1.25C_4(t-1) - 0.45C_4(t-2) + 0.15C_4(t-3) + 0.001M_6(t-3)$$

The premise variable membership functions A_1^{1f} to A_4^{3f} are shown in Fig.6.13

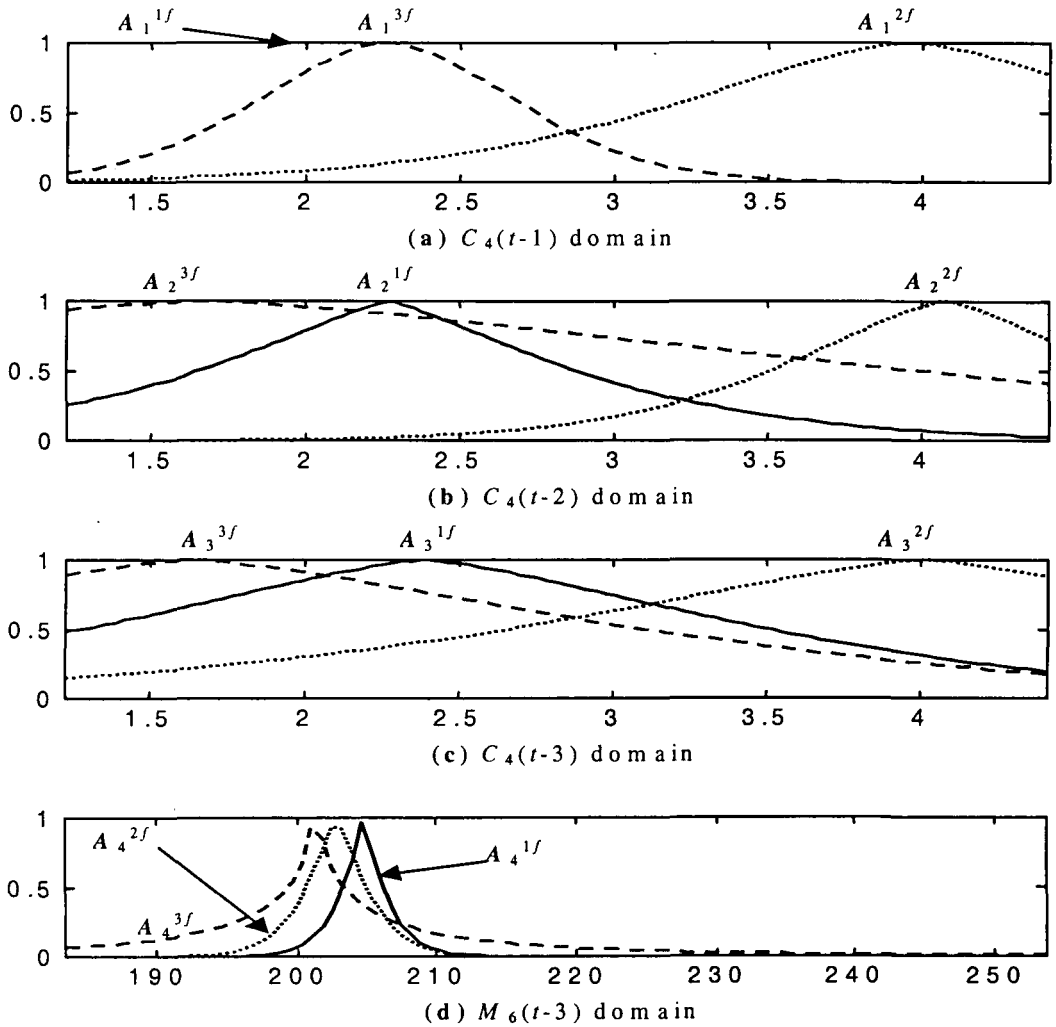


Fig. 6.13: Premise membership functions for GFM class II rules, of $C_4(t)$ variable

Since the membership value of A_1^{1f} remains at "1" over the entire domain of the variable $C_4(t\text{-}1)$, it can be removed from the R^{1f}. The modified form of fuzzy rule can be written as follows:

R^{1f} : if $C_4(t-2)$ is $A_2^{1f} \wedge C_4(t-3)$ is $A_3^{1f} \wedge M_6(t-3)$ is A_4^{1f} then $C_4(t)$ is $B^{1f}\left(f^{1f}(x), 3.23\right)$;

R^{2f} : if $C_4(t-1)$ is $A_1^{2f} \wedge C_4(t-2)$ is $A_2^{2f} \wedge C_4(t-3)$ is $A_3^{2f} \wedge M_6(t-3)$ is A_4^{2f} then $C_4(t)$ is $B^{2f}\left(f^{2f}(x), 3.29\right)$;

R^{3f} : if $C_4(t-1)$ is $A_1^{3f} \wedge C_4(t-2)$ is $A_2^{3f} \wedge C_4(t-3)$ is $A_3^{3f} \wedge M_6(t-3)$ is A_4^{3f} then $C_4(t)$ is $B^{3f}\left(f^{3f}(x), 3.28\right)$;

The output of the model and the actual values are plotted in Fig.6.14 (a). Figure 6.14(b) shows the model error in per unit.

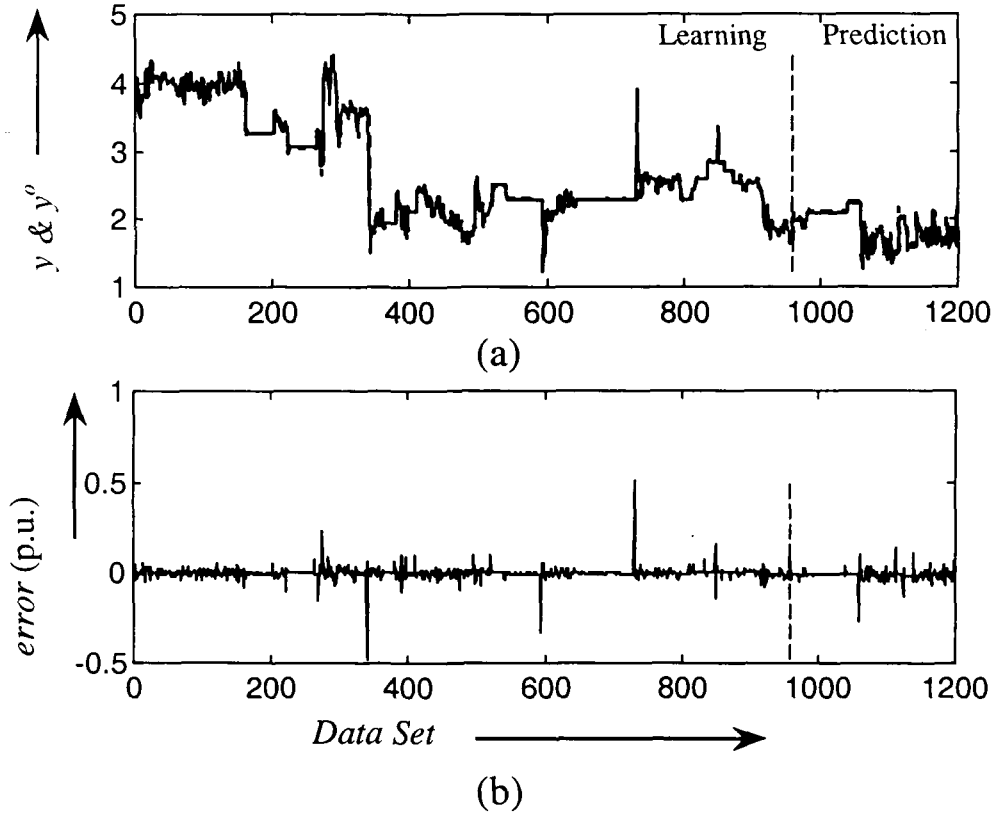


Fig.6.14: Plots of (a) y (_____, solid line) and y^o (- - - - - -,dashed line) (b) *model error* for class II GFM of $C_4(t)$.

Main Fractionator Auxiliary Reflux, $C_5(t)$

This variable is associated with fractionator unit and it depends on the $M_{10}(t\text{-}5)$, $C_5(t\text{-}1)$, $C_5(t\text{-}2)$, and $C_8(t)$. The model for Main Fractionator Auxiliary Reflux, i.e., $C_5(t)$ have second order dynamics instant effect of Column Bottom Circulation, i.e., $C_8(t)$ and 50 minutes delayed effect of Fractionator Bottom Temperature, i.e., $M_{10}(t\text{-}5)$. The fuzzy rules are given as follows:

R^{1f} : if $M_{10}(t-5)$ is $A_1^{1f} \wedge C_5(t-1)$ is $A_2^{1f} \wedge C_5(t-2)$ is $A_3^{1f} \wedge C_8(t)$ is A_4^{1f} then $C_5(t)$ is $B^{1f}\left(f^{1f}(x), 647.50\right)$;

R^{2f} : if $M_{10}(t-5)$ is $A_1^{2f} \wedge C_5(t-1)$ is $A_2^{2f} \wedge C_5(t-2)$ is $A_3^{2f} \wedge C_8(t)$ is A_4^{2f} then $C_5(t)$ is $B^{2f}\left(f^{2f}(x), 647.50\right)$;

R^{3f} : if $M_{10}(t-5)$ is $A_1^{3f} \wedge C_5(t-1)$ is $A_2^{3f} \wedge C_5(t-2)$ is $A_3^{3f} \wedge C_8(t)$ is A_4^{3f} then $C_5(t)$ is $B^{3f}\left(f^{3f}(x), 647.50\right)$;

where,

$$f^{1f}(x) = -1877.40 + 7.16M_{10}(t-5) - 1.98C_5(t-1) - 1.62C_5(t-2) + 0.21C_8(t)$$

$$f^{2f}(x) = 2779.70 - 7.85M_{10}(t-5) + 4.31C_5(t-1) - 0.60C_5(t-2) + 0.17C_8(t)$$

$$f^{3f}(x) = -276.2 - 2.49M_{10}(t-5) + 3.20C_5(t-1) - 0.79C_5(t-2) + 0.27C_8(t)$$

The premise variable membership functions A_1^{1f} to A_4^{3f} are shown in Fig.6.15

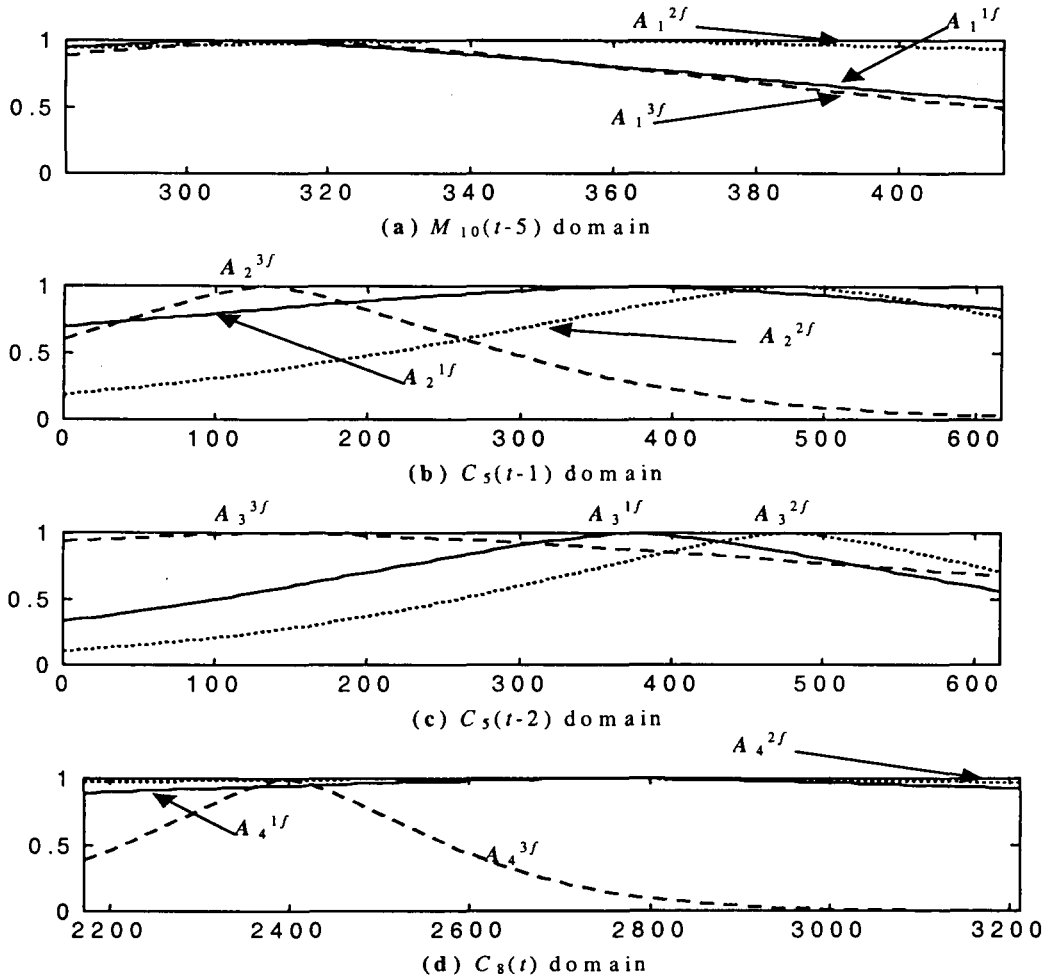


Fig. 6.15: Premise membership functions for class II GFM rules, of $C_5(t)$ variable

Since the membership value of A_1^{2f} and A_4^{1f} & A_4^{2f} remains at "1" over the entire domain of the variable M_{10} (t-5) and $C_8(t)$, the variable $M_{10}(t\text{-}5)$ from the R^{2f} and

the variable $C_8(t)$ from the R^{1f} & R^{2f} can be removed. The modified form of fuzzy rule can be written as follows:

R^{1f} : if $M_{10}(t-5)$ is $A_1^{1f} \wedge C_5(t-1)$ is $A_2^{1f} \wedge C_5(t-2)$ is A_3^{1f} then $C_5(t)$ is $B^{1f}\left(f^{1f}(x), 647.50\right)$;

R^{2f} : if $C_5(t-1)$ is $A_2^{2f} \wedge C_5(t-2)$ is A_3^{2f} then $C_5(t)$ is $B^{2f}\left(f^{2f}(x), 647.50\right)$;

R^{3f} : if $M_{10}(t-5)$ is $A_1^{3f} \wedge C_5(t-1)$ is $A_2^{3f} \wedge C_5(t-2)$ is $A_3^{3f} \wedge C_8(t)$ is A_4^{3f} then $C_5(t)$ is $B^{3f}\left(f^{3f}(x), 647.50\right)$;

The output of the model and the actual values are plotted in Fig.6.16 (a). Figure 6.16(b) shows the model error in per unit.

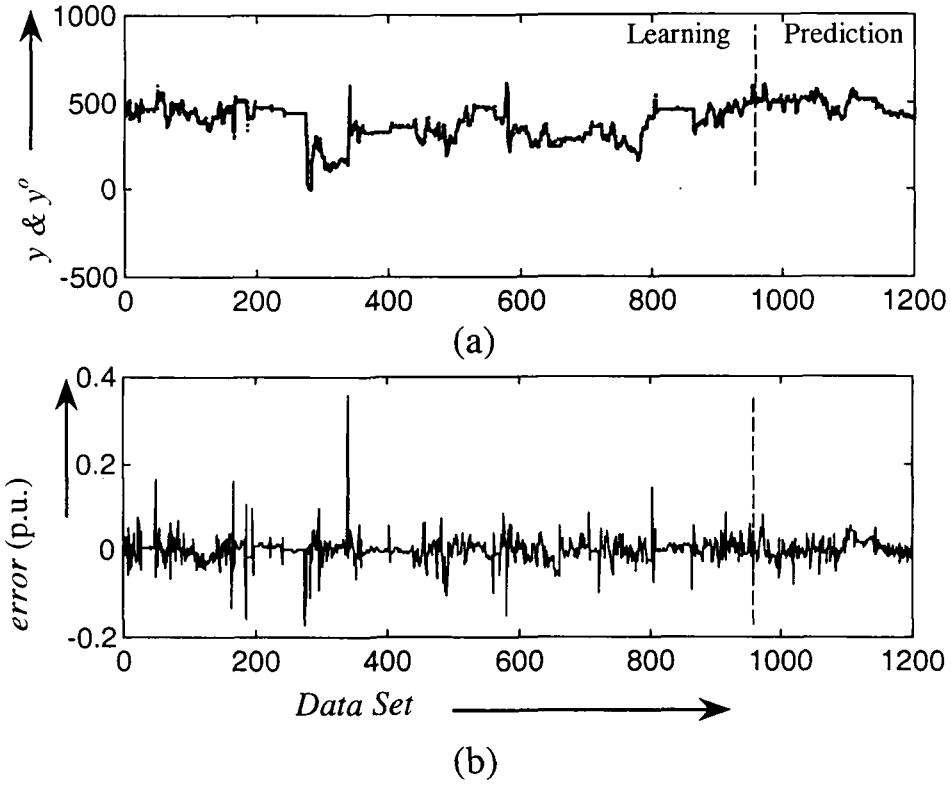


Fig.6.16: Plots of (a) y (_____, solid line) and y^o (- - - - - -,dashed line) (b) *model error* for class II GFM of $C_5(t)$.

Main Reflux, C_6 (t)

This variable is associated with fractionator unit and it depends on the variable C6 (t-1). The variable $C_6(t)$ exhibits the first order dynamics. The fuzzy rules are given as follows:

R^{1f} : if $C_6(t-1)$ is A_1^{1f} then $C_6(t)$ is $B^{1f}\left(f^{1f}(x), 204.02\right)$;

R^{2f} : if $C_6(t-1)$ is A_1^{2f} then $C_6(t)$ is $B^{2f}\left(f^{2f}(x), 204.02\right)$;

where,

$$f^{1f}(x) = -112.94 + 1.06C_6(t-1)$$

$$f^{2f}(x) = 65.59 + 0.97C_6(t-1)$$

The premise variable membership functions A_1^{1f} & A_1^{2f} are shown in Fig.6.17

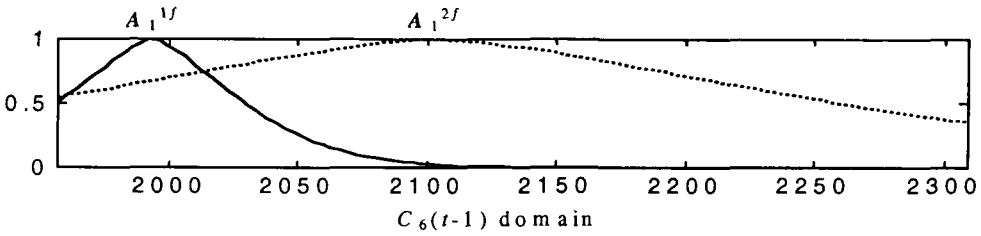


Fig. 6.17: Premise membership functions for class II GFM rules, of $C_6(t)$ variable

The output of the model and the actual values are plotted in Fig.6.18 (a). Figure 6.18(b) shows the model error in per unit.

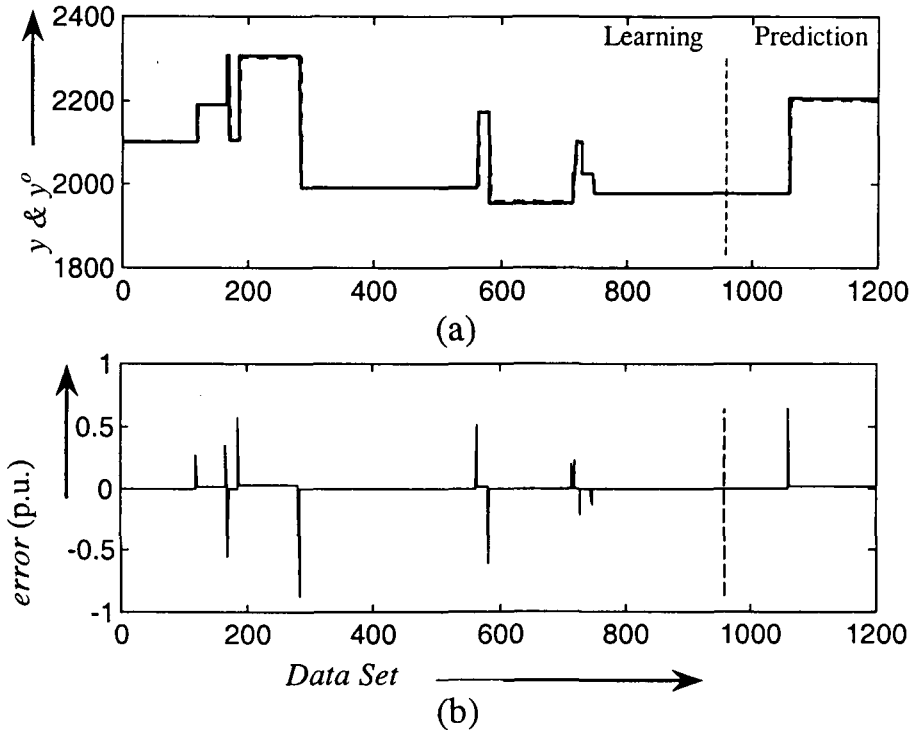


Fig.6.18: Plots of (a) y (_____, solid line) and y^o (- - - - - -,dashed line) (b) *model error* for class II GFM of $C_6(t)$

***Column Bottom Circulation I*, $C_7(t)$**

This variable is associated with fractionator unit and it depends on $C_7(t\text{-}1)$, $M_5(t\text{-}3)$, and $M_5(t\text{-}2)$. The variable $C_7(t)$,has the first order dynamics with at least 20 minutes delayed effect of catalytic Circulation rate, i.e., M_5. The fuzzy rules are given as follows:

R^{1f} :if $C_7(t-1)$ is $A_1^{1f} \wedge M_5(t-3)$ is $A_2^{1f} \wedge M_5(t-2)$ is A_3^{1f} then $C_7(t)$ is $B^{1f}\left(f^{1f}(x), 36.80\right)$;

R^{2f} :if $C_7(t-1)$ is $A_1^{2f} \wedge M_5(t-3)$ is $A_2^{2f} \wedge M_5(t-2)$ is A_3^{2f} then $C_7(t)$ is $B^{2f}\left(f^{2f}(x), 36.80\right)$;

R^{3f} :if $C_7(t-1)$ is $A_1^{3f} \wedge M_5(t-3)$ is $A_2^{3f} \wedge M_5(t-2)$ is A_3^{3f} then $C_7(t)$ is $B^{3f}\left(f^{3f}(x), 36.80\right)$;

where,

$$f^{1f}(x) = -12756.00 + 10.78C_7(t-1) - 41.12M_5(t-3) - 30.06M_5(t-2)$$

$$f^{2f}(x) = -12874.00 + 10.41C_7(t-1) + 5.92M_5(t-3) + 11.17M_5(t-2)$$

$$f^{3f}(x) = 20231.00 - 14.30C_7(t-1) + 35.94M_5(t-3) + 16.16M_5(t-2)$$

The premise variable membership functions A_1^{1f} to A_3^{3f} are shown in Fig.6.19

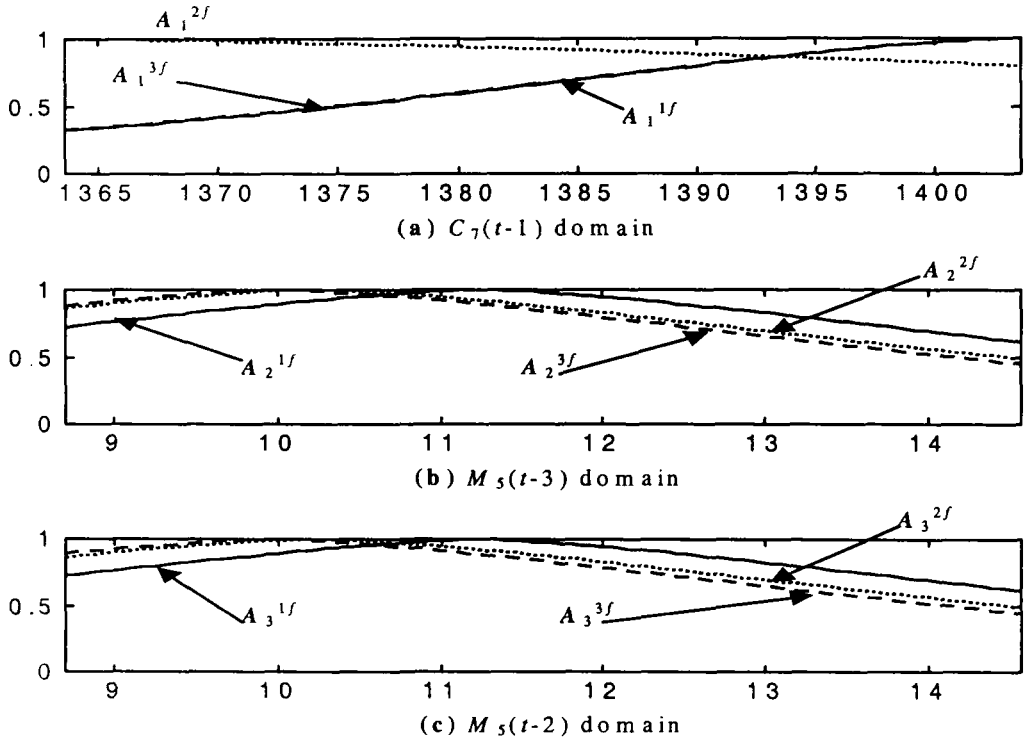


Fig. 6.19: Premise membership functions for class II GFM rules, of $C_7(t)$ variable

The output of the model and the actual values are plotted in Fig.6.20 (a). Figure 6.20(b) shows the model error in per unit.

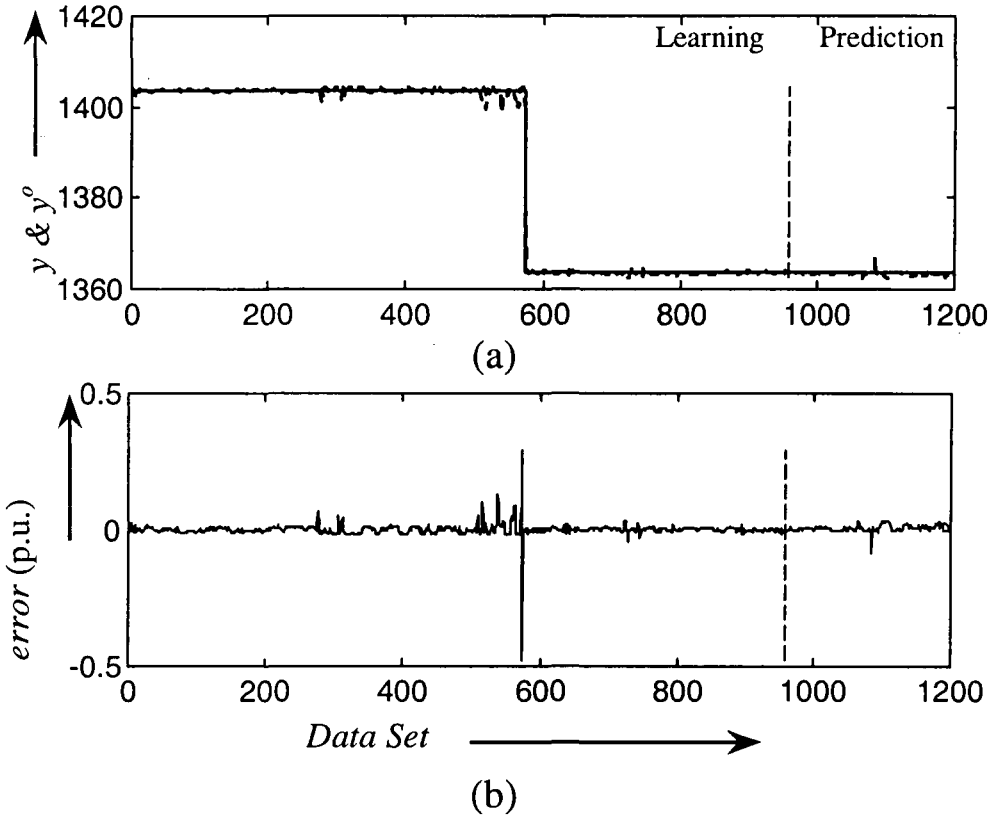


Fig.6.20: Plots of (a) y (_____, solid line) and y^o (- - - - - -,dashed line) (b) *model error* for class II GFM of $C_7(t)$.

Column Bottom Circulation II, $C_8(t)$

This variable is associated with fractionator unit and it depends on $C_8(t\text{-}1)$, $C_5(t)$, $M_6(t\text{-}5)$, $M_6(t\text{-}3)$, and $M_9(t\text{-}5)$. The variable $C_8(t)$ has the first order dynamics with at least 30 minutes delayed effect of Compressor Amperage, instantaneous effect of the variable Main Fractionator Auxiliary Reflux and 50 minutes delayed effect of Side Draw off Temperature for HCO. The fuzzy rules are given as follows:

R^{1f} :if $C_8(t-1)$ is $A_1^{1f} \wedge C_5(t)$ is $A_2^{1f} \wedge M_6(t-5)$ is $A_3^{1f} \wedge M_6(t-3)$ is $A_4^{1f} \wedge M_9(t-5)$ is A_5^{1f} then $C_8(t)$ is $B^{1f}\left(f^{1f}(x), 1.02\times10^3\right)$;

R^{2f} :if $C_8(t-1)$ is $A_1^{2f} \wedge C_5(t)$ is $A_2^{2f} \wedge M_6(t-5)$ is $A_3^{2f} \wedge M_6(t-3)$ is $A_4^{2f} \wedge M_9(t-5)$ is A_5^{2f} then $C_8(t)$ is $B^{2f}\left(f^{2f}(x), 1.02\times10^3\right)$;

R^{3f} :if $C_8(t-1)$ is $A_1^{3f} \wedge C_5(t)$ is $A_2^{3f} \wedge M_6(t-5)$ is $A_3^{3f} \wedge M_6(t-3)$ is $A_4^{3f} \wedge M_9(t-5)$ is A_5^{3f} then $C_8(t)$ is $B^{3f}\left(f^{3f}(x), 1.02\times10^3\right)$;

where,

$$f^{1f}(x) = -7164.10 + 5.90C_8(t-1) - 2.86C_5(t) - 6.01M_6(t-5) - 43.67M_6(t-3) + 10.30M_9(t-5)$$

$$f^{2f}(x) = 6083.90 + 0.77C_8(t-1) + 3.66C_5(t) - 14.02M_6(t-5) + 5.36M_6(t-3) - 5.09M_9(t-5)$$

$$f^{3f}(x) = 2862.40 - 3.00C_8(t-1) - 1.82C_5(t) + 19.33M_6(t-5) + 31.96M_6(t-3) - 13.99M_9(t-5)$$

The premise variable membership functions A_1^{1f} to A_5^{3f} are shown in Fig.6.21

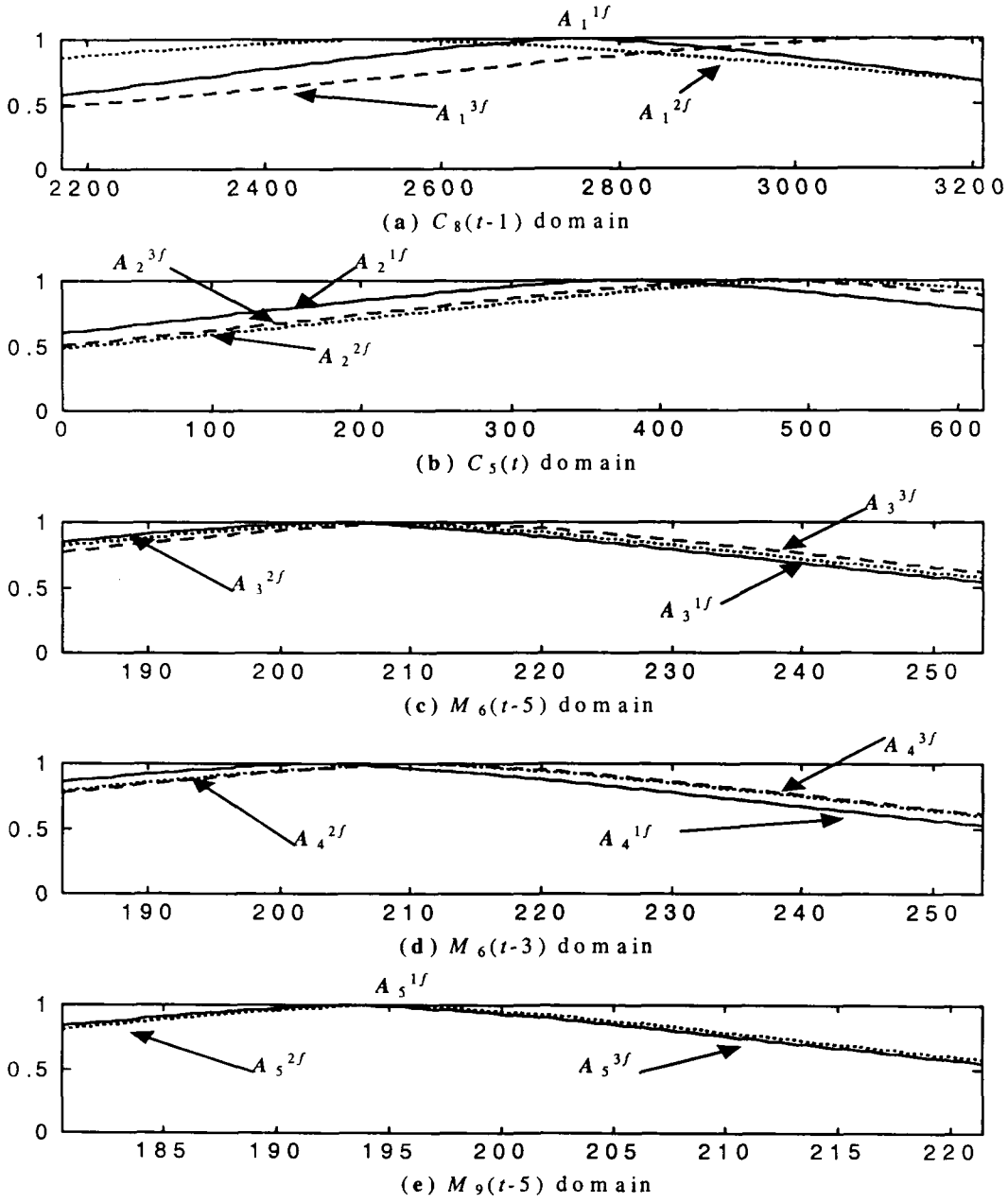


Fig. 6.21: Premise membership functions for class II GFM rules, of $C_8(t)$ variable.

The output of the model and the actual values are plotted in Fig.6.22 (a). Figure 6.22(b) shows the model error in per unit.

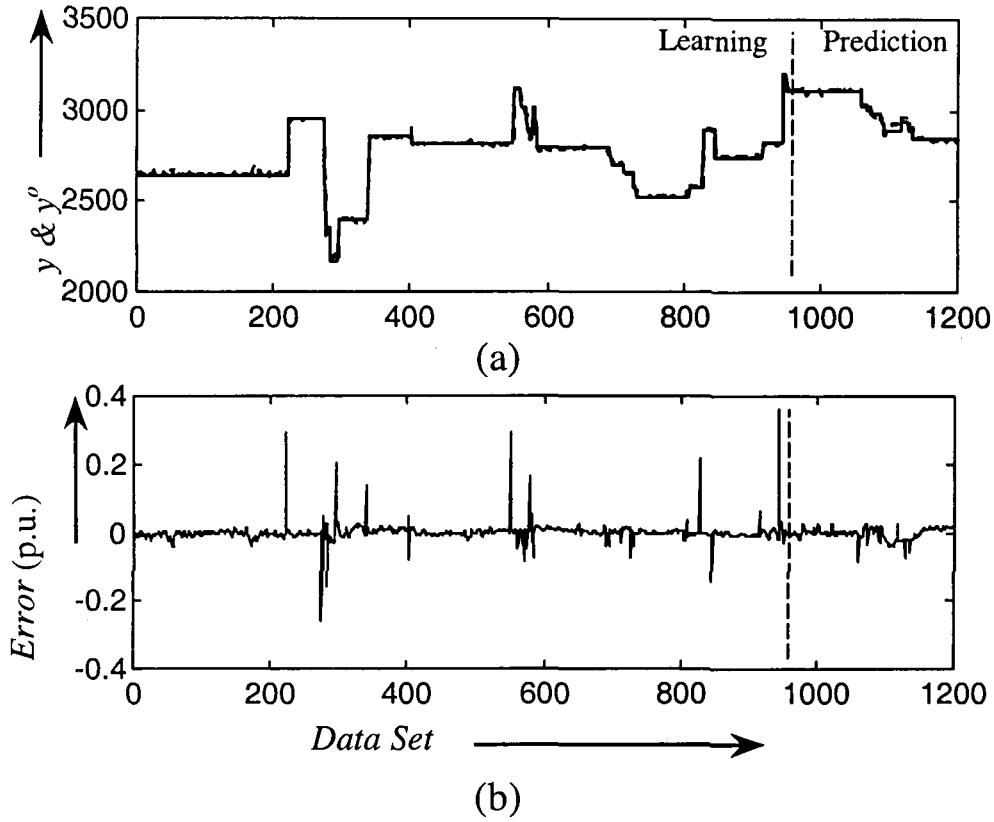


Fig.6.22: Plots of (a) y (————, solid line) and y^o (- - - - - -,dashed line) (b) *model error* for class II GFM of $C_8(t)$.

Clarified Oil Production, $C_9(t)$

This variable is associated with fractionator unit and it depends on $C_9(t\text{-}1)$, $C_9(t\text{-}2)$ and $M_1(t\text{-}2)$. The variable $C_9(t)$ possesses the second order dynamics and 20 minutes delayed effect of Fractionator Bottom Level, i.e., $M_1(t\text{-}2)$. The fuzzy rules are given as follows:

$$R^{1f}: \text{if } C_9(t-1) \text{ is } A_1^{1f} \wedge C_9(t-2) \text{ is } A_2^{1f} \wedge M_1(t-2) \text{ is } A_3^{1f} \quad \text{then } C_9(t) \text{ is } B^{1f}\left(f^{1f}(x), 392.44\right);$$

$$R^{2f}: \text{if } C_9(t-1) \text{ is } A_1^{2f} \wedge C_9(t-2) \text{ is } A_2^{2f} \wedge M_1(t-2) \text{ is } A_3^{2f} \quad \text{then } C_9(t) \text{ is } B^{2f}\left(f^{2f}(x), 392.44\right);$$

$$R^{3f}: \text{if } C_9(t-1) \text{ is } A_1^{3f} \wedge C_9(t-2) \text{ is } A_2^{3f} \wedge M_1(t-2) \text{ is } A_3^{3f} \quad \text{then } C_9(t) \text{ is } B^{3f}\left(f^{3f}(x), 392.44\right);$$

where,

$$f^{1f}(x) = 38.98 + 1.33C_9(t-1) - 0.48C_9(t-2) - 0.07M_1(t-2)$$

$$f^{2f}(x) = -26.95 + 1.38C_9(t-1) - 0.30C_9(t-2) + 0.92M_1(t-2)$$

$$f^{3f}(x) = -118.33 + 1.18C_9(t-1) - 0.08C_9(t-2) + 0.72M_1(t-2)$$

The premise variable membership functions A_1^{1f} to A_3^{3f} are shown in Fig.6.23

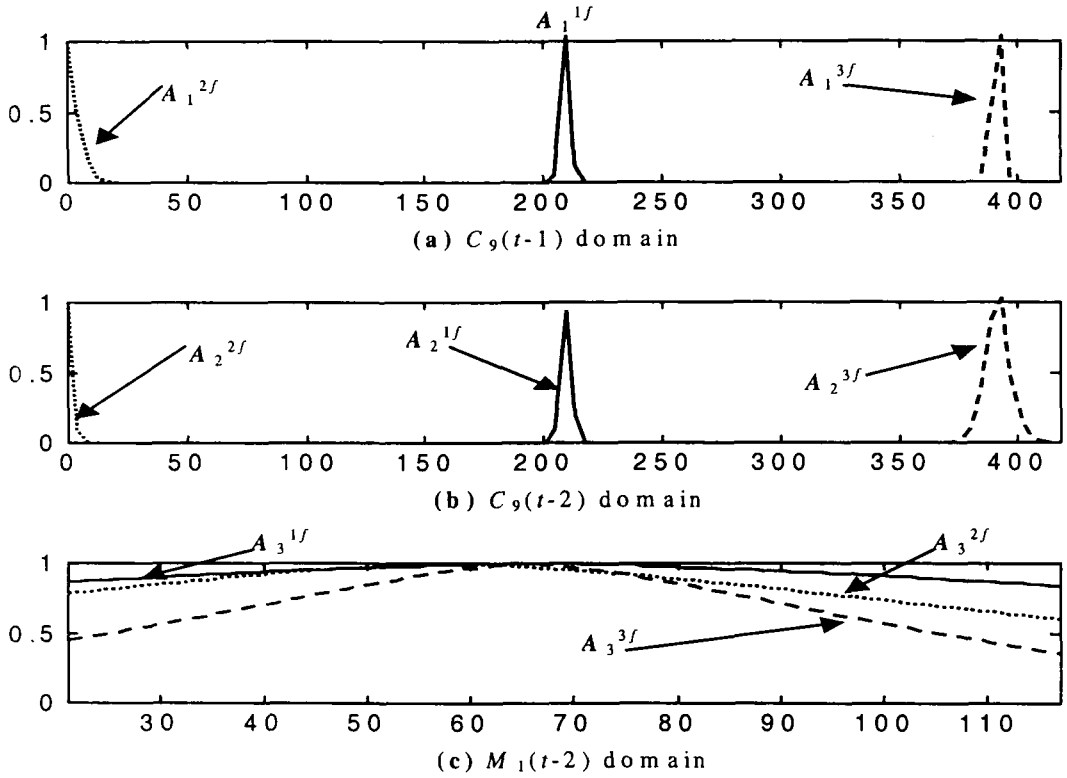


Fig. 6.23: Premise membership functions for class II GFM rules, of $C_9(t)$ variable

The output of the model and the actual values are plotted in Fig.6.24(a). Figure 6.24(b) shows the model error in per unit.

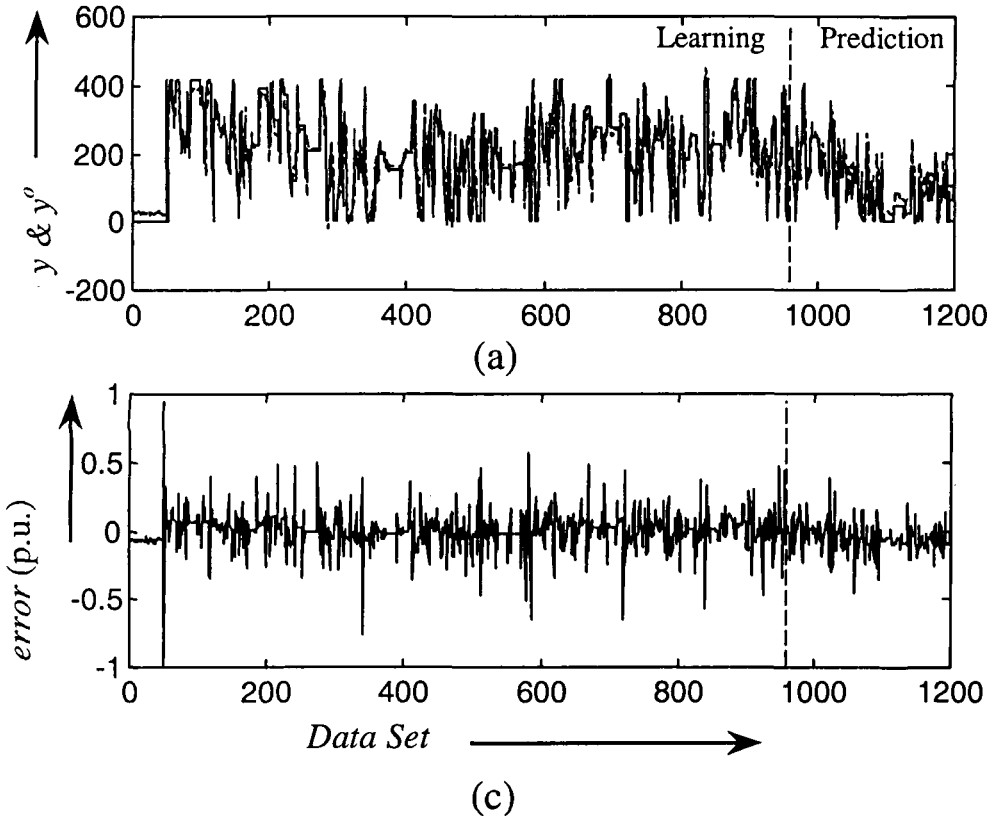


Fig.6.24: Plots of (a) y (_____, solid line) and y^o (- - - - - -,dashed line) (b) *model error* for class II GFM of $C_9(t)$.

Cold feed to Reactor, $C_{10}(t)$

This variable is associated with RRS unit and it depends on $C_{10}(t\text{-}1)$, $M_6(t)$, and $M_6(t\text{-}1)$. The variable $C_{10}(t)$ has the second order dynamics with at least the instantaneous effect Compressor Amperage. The fuzzy rules are given as follows:

R^{1f} :if $C_{10}(t-1)$ is $A_1^{1f} \wedge M_6(t)$ is $A_2^{1f} \wedge M_6(t-1)$ is A_3^{1f} then $C_{10}(t)$ is $B^{1f}\left(f^{1f}(x), 851.08\right)$;

R^{2f} :if $C_{10}(t-1)$ is $A_1^{2f} \wedge M_6(t)$ is $A_2^{2f} \wedge M_6(t-1)$ is A_3^{2f} then $C_{10}(t)$ is $B^{2f}\left(f^{2f}(x), 851.08\right)$;

R^{3f} :if $C_{10}(t-1)$ is $A_1^{3f} \wedge M_6(t)$ is $A_2^{3f} \wedge M_6(t-1)$ is A_3^{3f} then $C_{10}(t)$ is $B^{3f}\left(f^{3f}(x), 851.08\right)$;

R^{4f} :if $C_{10}(t-1)$ is $A_1^{4f} \wedge M_6(t)$ is $A_2^{4f} \wedge M_6(t-1)$ is A_3^{4f} then $C_{10}(t)$ is $B^{4f}\left(f^{4f}(x), 873.48\right)$;

where,

$$f^{1f}(x) = -2933.30 + 1.06C_{10}(t-1) - 0.35M_6(t) + 15.32M_6(t-1)$$

$$f^{2f}(x) = 5885.80 + 1.33C_{10}(t-1) - 4.34M_6(t) - 24.49M_6(t-1)$$

$$f^{3f}(x) = -3183.80 + 2.27C_{10}(t-1) - 1.72M_6(t) + 9.47M_6(t-1)$$

$$f^{4f}(x) = -2004.50 + 0.84C_{10}(t-1) + 1.60M_6(t) + 6.53M_6(t-1)$$

The premise variable membership functions A_1^{1f} to A_3^{4f} are shown in Fig.6.25

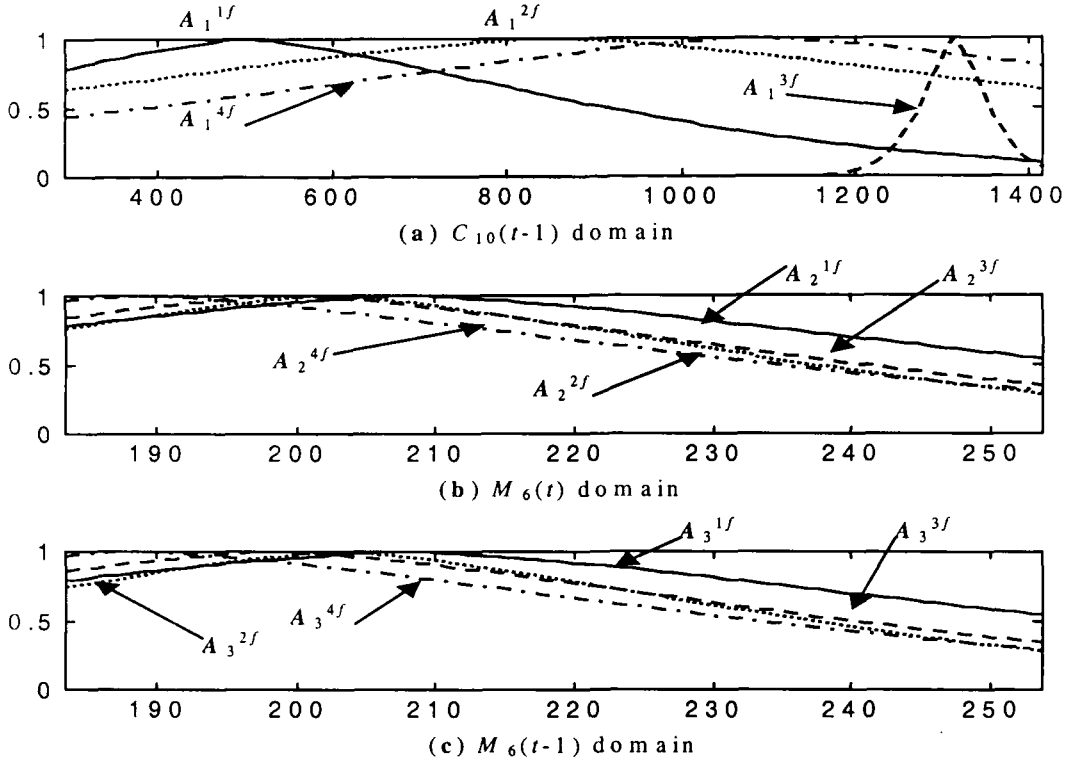


Fig. 6.25: Premise membership functions for class II GFM rules, of $C_{10}(t)$ variable.

The output of the model and the actual values are plotted in Fig.6.26 (a). Figure 6.26(b) shows the model error in per unit.

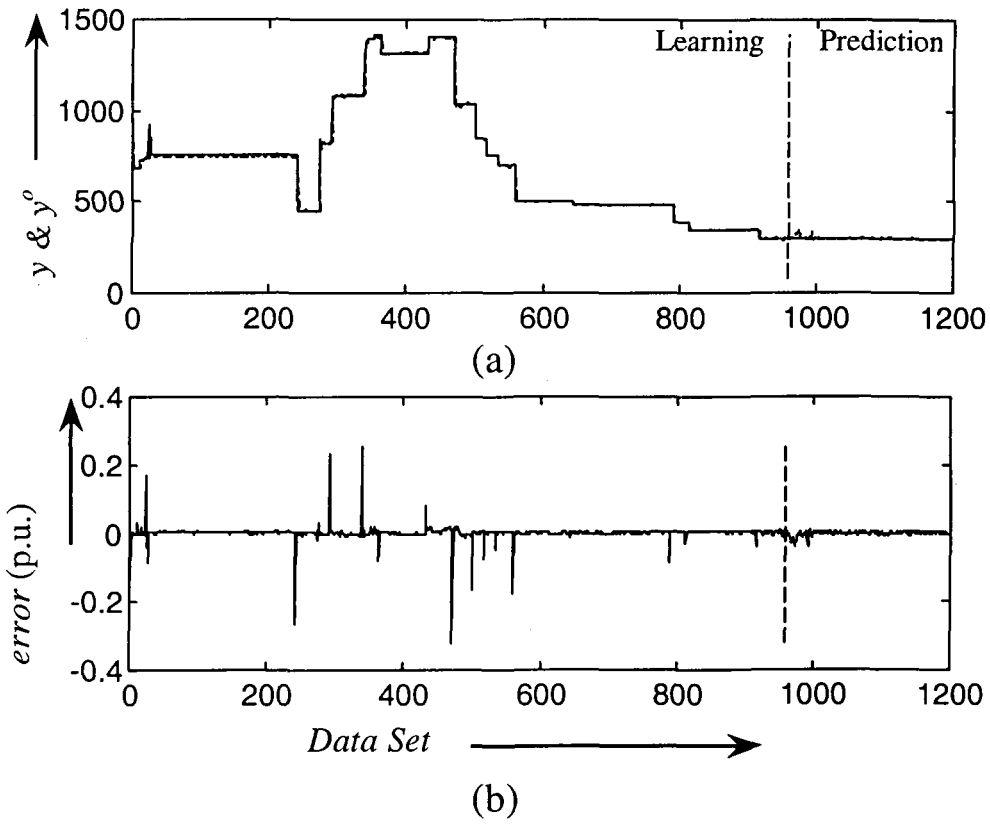


Fig.6.26: Plots of (a) y (_____, solid line) and y^o (- - - - - -,dashed line) (b) *model error* for class II GFM of $C_{10}(t)$.

Fractionator Bottom Level, $M_1(t)$

This variable is associated with the fractionator unit and it depends on $M_1(t\text{-}1)$, $M_1(t\text{-}2)$, $M_6(t\text{-}2)$ and $M_6(t)$. The variable $M_1(t)$ has the second order dynamics with at least the instantaneous effect of compressor amperage. The fuzzy rules are given as follows:

$$R^{1f}: \text{if } M_1(t-1) \text{ is } A_1^{1f} \wedge M_1(t-2) \text{ is } A_2^{1f} \wedge M_6(t-2) \text{ is } A_3^{1f} \wedge M_6(t) \text{ is } A_4^{1f} \quad \text{then } M_1(t) \text{ is } B^{1f}\left(f^{1f}(x), 83.37\right);$$

$$R^{2f}: \text{if } M_1(t-1) \text{ is } A_1^{2f} \wedge M_1(t-2) \text{ is } A_2^{2f} \wedge M_6(t-2) \text{ is } A_3^{2f} \wedge M_6(t) \text{ is } A_4^{2f} \quad \text{then } M_1(t) \text{ is } B^{2f}\left(f^{2f}(x), 83.37\right);$$

$$R^{3f}: \text{if } M_1(t-1) \text{ is } A_1^{3f} \wedge M_1(t-2) \text{ is } A_2^{3f} \wedge M_6(t-2) \text{ is } A_3^{3f} \wedge M_6(t) \text{ is } A_4^{3f} \quad \text{then } M_1(t) \text{ is } B^{3f}\left(f^{3f}(x), 83.37\right);$$

where,

$$f^{1f}(x) = 21.12 + 1.95M_1(t-1) - 1.07M_1(t-2) - 0.09M_6(t-2) + 0.03M_6(t)$$

$$f^{2f}(x) = 28.12 + 1.41M_1(t-1) - 0.59M_1(t-2) + 0.79M_6(t-2) - 0.87M_6(t)$$

$$f^{3f}(x) = 45.06 + 0.32M_1(t-1) - 0.13M_1(t-2) + 0.98M_6(t-2) - 0.97M_6(t)$$

The premise variable membership functions A_1^{1f} to A_4^{3f} are shown in Fig.6.27

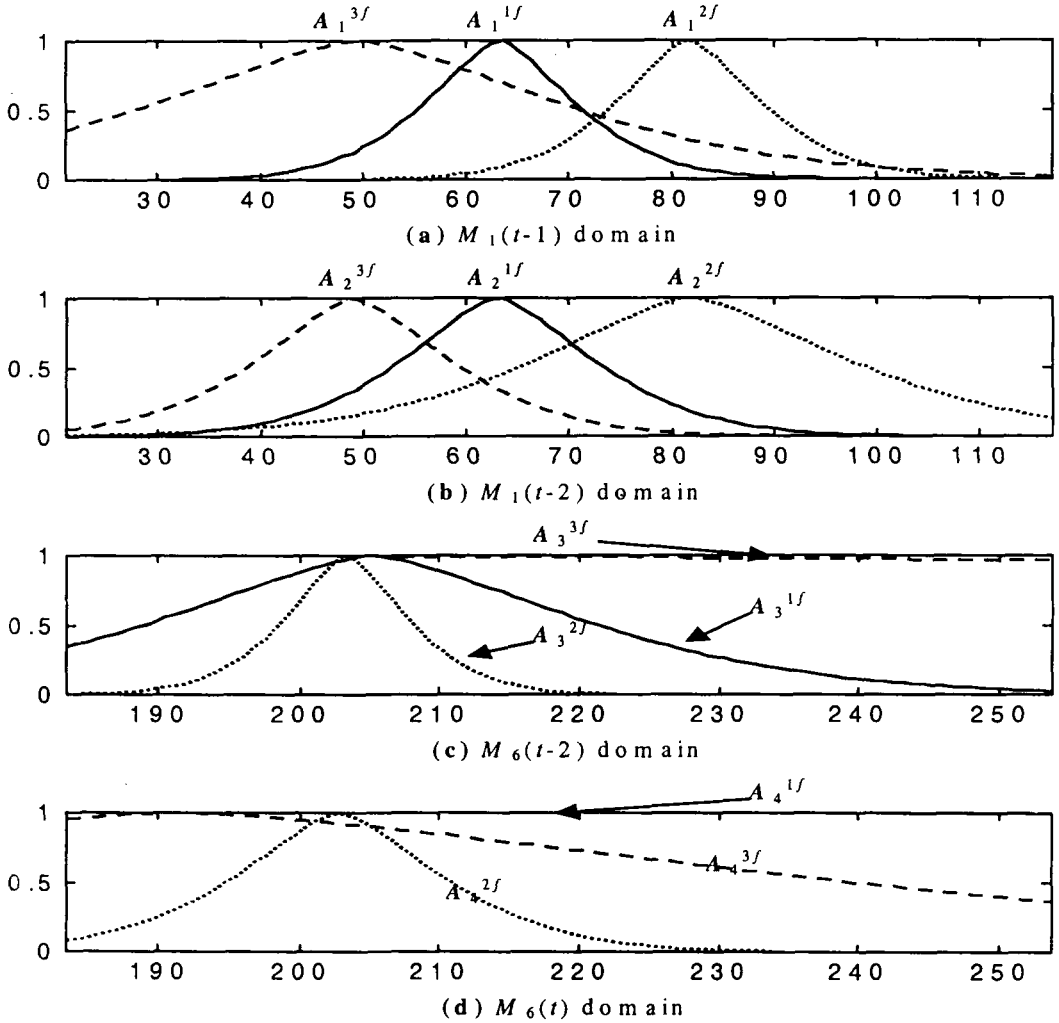


Fig. 6.27: Premise membership functions for class II GFM rules, of $M_1(t)$ variable

Since the membership value of and A_3^{3f} & A_4^{1f} remains at "1" over the entire domain of the variable $M_6(t\text{-}2)$ and $M_6(t)$, the variable $M_6(t\text{-}2)$ from the R^{3f} and the variable $M_6(t)$ from the R^{1f} can be removed. The modified form of fuzzy rule can be written as follows:

R^{1f} : if $M_1(t-1)$ is $A_1^{1f} \wedge M_1(t-2)$ is $A_2^{1f} \wedge M_6(t-2)$ is A_3^{1f} then $M_1(t)$ is $B^{1f}\left(f^{1f}(x), 83.37\right)$;

R^{2f} : if $M_1(t-1)$ is $A_1^{2f} \wedge M_1(t-2)$ is $A_2^{2f} \wedge M_6(t-2)$ is $A_3^{2f} \wedge M_6(t)$ is A_4^{2f} then $M_1(t)$ is $B^{2f}\left(f^{2f}(x), 83.37\right)$;

R^{3f} : if $M_1(t-1)$ is $A_1^{3f} \wedge M_1(t-2)$ is $A_2^{3f} \wedge M_6(t)$ is A_4^{3f} then $M_1(t)$ is $B^{3f}\left(f^{3f}(x), 83.37\right)$;

The output of the model and the actual values are plotted in Fig.6.28(a). Figure 6.28(b) shows the model error in per unit.

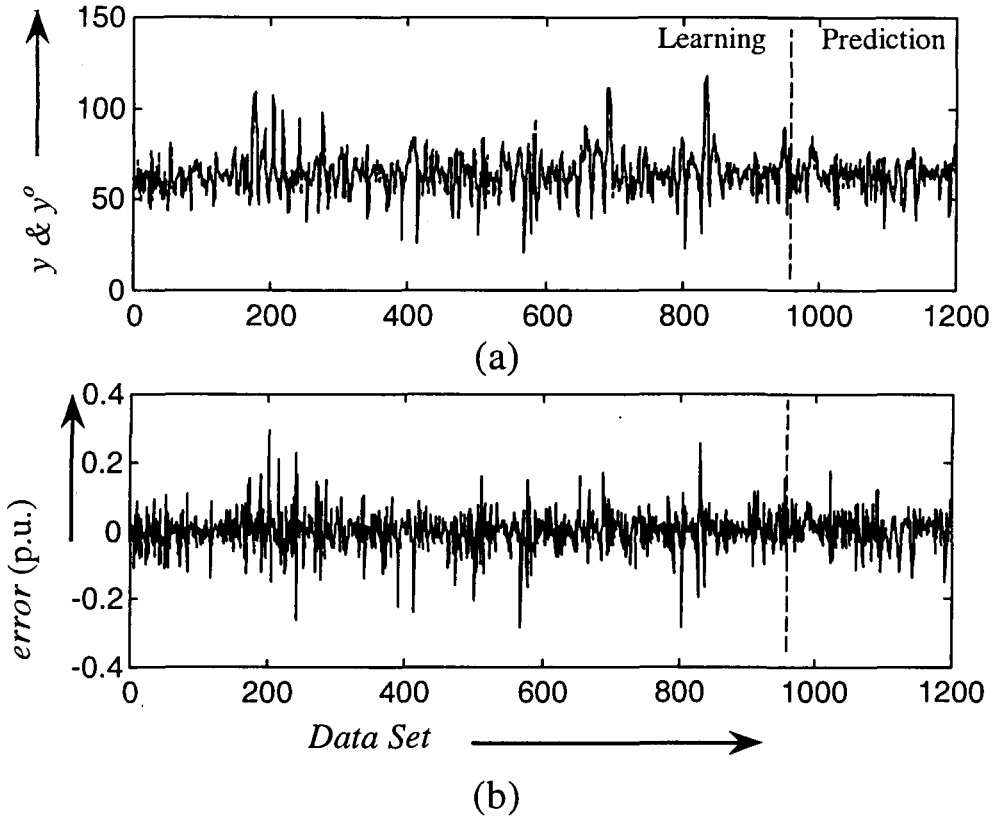


Fig.6.28: Plots of (a) y (————, solid line) and y^o (- - - - - -,dashed line) (b) *model error* for class II GFM of $M_1(t)$.

Regenerator Bed Temperature, $M_2(t)$

This variable is associated with the RRS unit and it depends on $M_3(t\text{-}4)$, $M_2(t\text{-}5)$, $M_2(t\text{-}3)$, $M_2(t\text{-}2)$, and $M_2(t\text{-}1)$. The variable $M_2(t)$ exhibits the fifth order dynamics and 40 minutes delayed effect of the Flue Gas Temperature, i.e., $M_3(t\text{-}4)$. The fuzzy rules are given as follows:

R^{1f} : if $M_3(t-4)$ is $A_1^{1f} \wedge M_2(t-5)$ is $A_2^{1f} \wedge M_2(t-3)$ is A_3^{1f}
$\wedge M_2(t-2)$ is $A_4^{1f} \wedge M_2(t-1)$ is A_5^{1f} then $M_2(t)$ is $B^{1f}\left(f^{1f}(x), 107.32\right)$

R^{2f} : if $M_3(t-4)$ is $A_1^{2f} \wedge M_2(t-5)$ is $A_2^{2f} \wedge M_2(t-3)$ is A_3^{2f}
$\wedge M_2(t-2)$ is $A_4^{2f} \wedge M_2(t-1)$ is A_5^{2f} then $M_2(t)$ is $B^{2f}\left(f^{2f}(x), 107.32\right)$

$$R^{3f}: \text{if } M_3(t-4) \text{ is } A_1^{3f} \wedge M_2(t-5) \text{ is } A_2^{3f} \wedge M_2(t-3) \text{ is } A_3^{3f} \wedge M_2(t-2) \text{ is } A_4^{3f} \wedge M_2(t-1) \text{ is } A_5^{3f} \quad \text{then } M_2(t) \text{ is } B^{3f}\left(f^{3f}(x), 107.32\right)$$

where,

$$f^{1f}(x) = -623.89 + 0.29M_3(t-4) - 0.24M_2(t-5) + 1.29M_2(t-3) - 2.54M_2(t-2) + 3.11M_2(t-1)$$

$$f^{2f}(x) = 506.93 - 0.80M_3(t-4) + 0.34M_2(t-5) - 1.18M_2(t-3) + 0.59M_2(t-2) + 1.31M_2(t-1)$$

$$f^{3f}(x) = 321.02 + 0.29M_3(t-4) - 0.06M_2(t-5) + 0.13M_2(t-3) + 0.28M_2(t-2) - 0.10M_2(t-1)$$

The premise variable membership functions A_1^{1f} to A_5^{3f} are shown in Fig.6.29

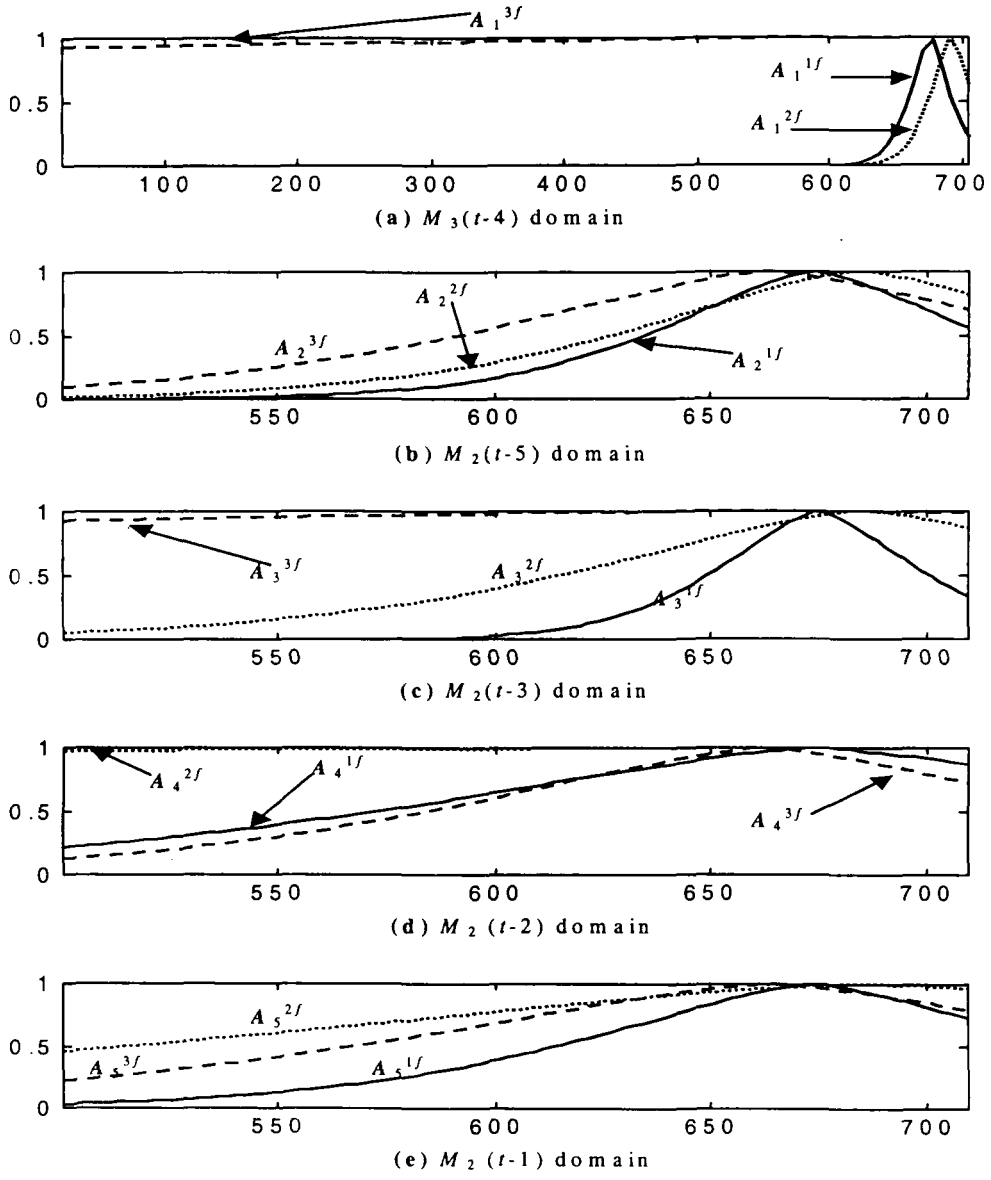


Fig. 6.29: Premise membership functions for class II GFM rules, of $M_2(t)$ variable

Since the membership value of A_1^{3f} and A_3^{3f} remains at "1" over the entire domain of the variable $M_3(t\text{-}4)$ and $M_2(t\text{-}3)$, these variable can be removed from R^{3f}. Moreover, A_4^{2f} also remains at "1" over the entire domain of the variable $M_2(t\text{-}2)$ and can be removed from R^{2f}. The modified form of fuzzy rule can be written as follows:

R^{1f} : if $M_3(t-4)$ is $A_1^{1f} \wedge M_2(t-5)$ is $A_2^{1f} \wedge M_2(t-3)$ is $A_3^{1f} \wedge M_2(t-2)$ is $A_4^{1f} \wedge M_2(t-1)$ is A_5^{1f} then $M_2(t)$ is $B^{1f}\left(f^{1f}(x), 107.32\right)$;

R^{2f} : if $M_3(t-4)$ is $A_1^{2f} \wedge M_2(t-5)$ is $A_2^{2f} \wedge M_2(t-3)$ is $A_3^{2f} \wedge M_2(t-1)$ is A_5^{2f} then $M_2(t)$ is $B^{2f}\left(f^{2f}(x), 107.32\right)$;

R^{3f} : if $M_2(t-5)$ is $A_2^{3f} \wedge M_2(t-2)$ is $A_4^{3f} \wedge M_2(t-1)$ is A_5^{3f} then $M_2(t)$ is $B^{3f}\left(f^{3f}(x), 107.32\right)$;

The output of the model and the actual values are plotted in Fig.6.30 (a). Figure 6.30(b) shows the model error in per unit.

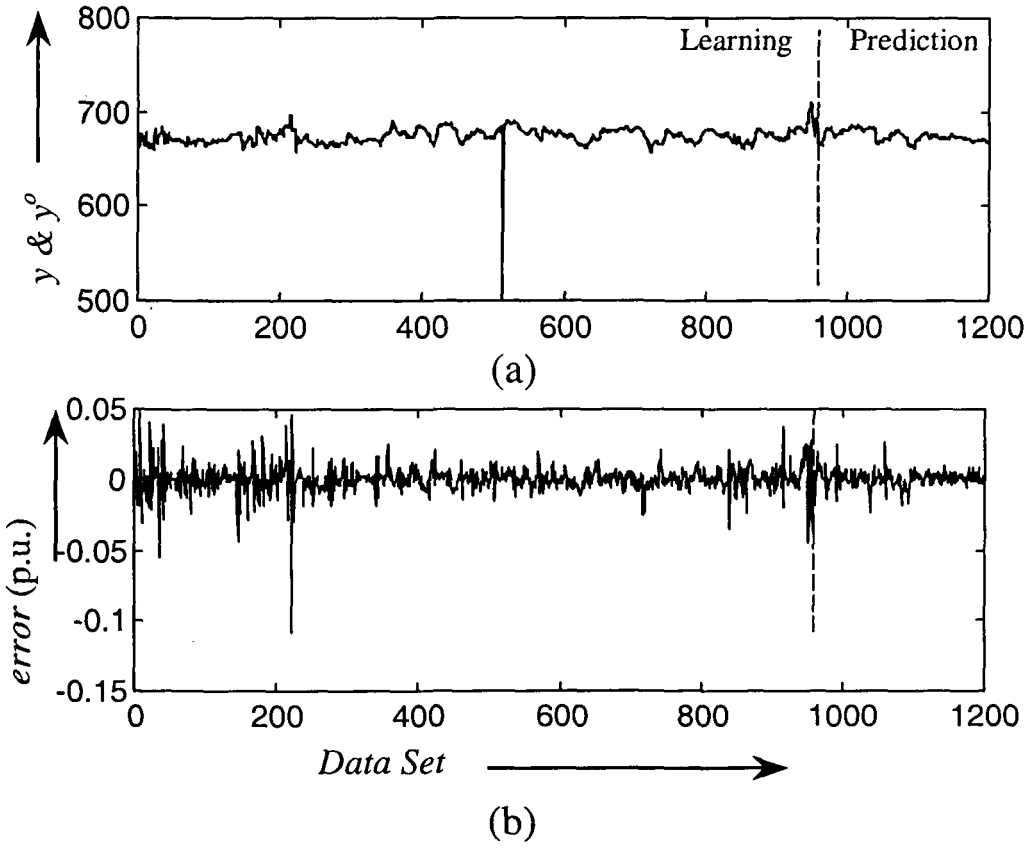


Fig.6.30: Plots of (a) y (————, solid line) and y^o (- - - - - -,dashed line) (b) *model error* for class II GFM of $M_2(t)$.

Flue Gas Temperature, $M_3(t)$

This variable is associated with RRS unit and it depends on $M_3(t\text{-}1)$. It has the first order dynamics. The fuzzy rules are given as follows:

$$R^{1f}: \text{if } M_3(t-1) \text{ is } A_1^{1f} \quad \text{then } M_3(t) \text{ is } B^{1f}\left(f^{1f}(x), 163.76\right);$$

$$R^{2f}: \text{if } M_3(t-1) \text{ is } A_1^{2f} \quad \text{then } M_3(t) \text{ is } B^{2f}\left(f^{2f}(x), 163.76\right);$$

$$R^{3f}: \text{if } M_3(t-1) \text{ is } A_1^{3f} \quad \text{then } M_3(t) \text{ is } B^{3f}\left(f^{3f}(x), 163.76\right);$$

where,

$$f^{1f}(x) = 1530.8 - 1.27M_3(t-1)$$

$$f^{2f}(x) = 596.00 + 0.15M_3(t-1)$$

$$f^{2f}(x) = 596.00 - 0.15M_3(t-1)$$

The premise variable membership functions $A_1^{1f} - A_1^{3f}$ are shown in Fig.6.31

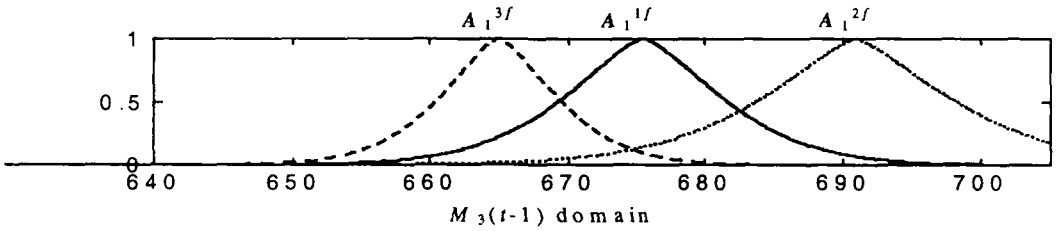


Fig. 6.31: Premise membership functions for class II GFM rules, of $M_3(t)$ variable

The output of the model and the actual values are plotted in Fig.6.32(a). Figure 6.32(b) shows the model error in per unit.

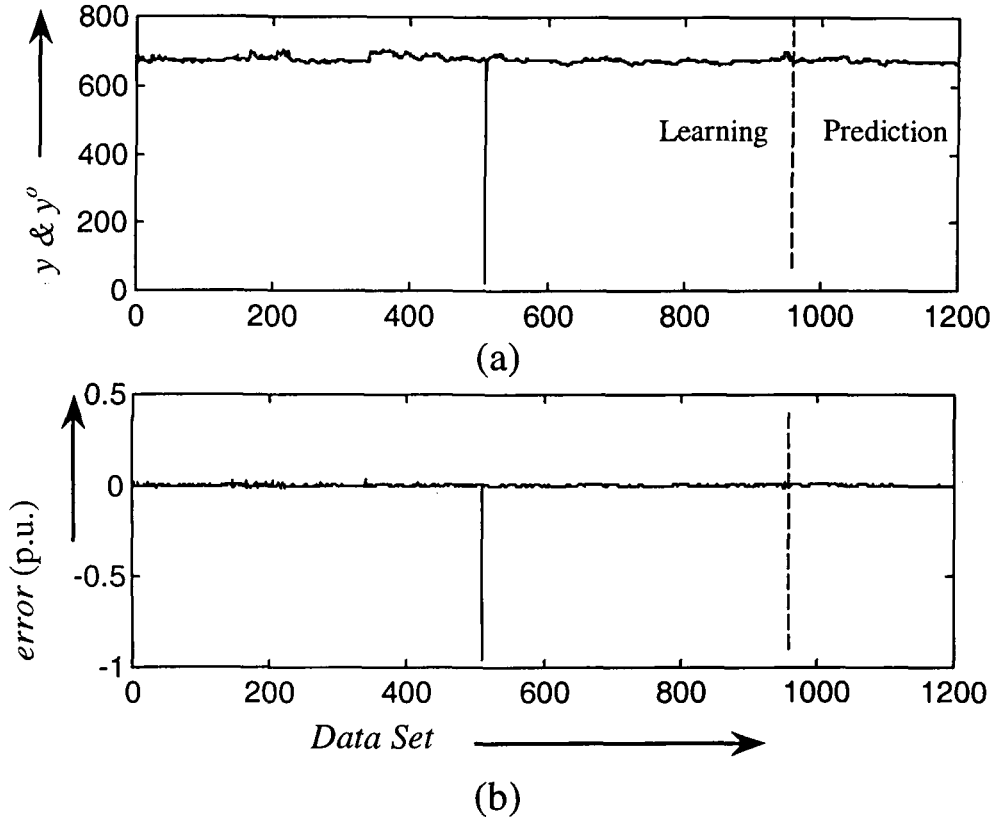


Fig.6.32: Plots of (a) y (————, solid line) and y^o (- - - - - -,dashed line) (b) *model error* for class II GFM of $M_3(t)$.

Reactor Bed Temperature, $M_4(t)$

This variable is associated with RRS unit and it depends on $M_{10}(t)$, $M_4(t\text{-}1)$, $M_6(t)$, $M_4(t\text{-}2)$, $M_6(t\text{-}2)$, $M_8(t)$, $M_8(t\text{-}1)$, and $M_8(t\text{-}4)$. The variable $M_4(t)$ possesses the second order dynamics with instantaneous effect of Compressor Amperage, Delta P across the Fractionator and Bottom vapor Temperature. The fuzzy rules are given as follows:

$$R^{1f}: \text{if } M_{10}(t) \text{ is } A_1^{1f} \wedge M_4(t-1) \text{ is } A_2^{1f} \wedge M_6(t) \text{ is } A_3^{1f} \wedge M_4(t-2) \text{ is } A_4^{1f} \wedge M_6(t-2) \text{ is } A_5^{1f} \wedge M_8(t) \text{ is } A_6^{1f} \wedge M_8(t-1) \text{ is } A_7^{1f} \wedge M_8(t-4) \text{ is } A_8^{1f} \text{ then } M_4(t) \text{ is } B^{1f}\left(f^{1f}(x), 24.58\right);$$

$$R^{2f}: \text{if } M_{10}(t) \text{ is } A_1^{2f} \wedge M_4(t-1) \text{ is } A_2^{2f} \wedge M_6(t) \text{ is } A_3^{2f} \wedge M_4(t-2) \text{ is } A_4^{2f} \wedge M_6(t-2) \text{ is } A_5^{2f} \wedge M_8(t) \text{ is } A_6^{2f} \wedge M_8(t-1) \text{ is } A_7^{2f} \wedge M_8(t-4) \text{ is } A_8^{2f} \text{ then } M_4(t) \text{ is } B^{2f}\left(f^{2f}(x), 24.58\right);$$

$$R^{3f}: \text{if } M_{10}(t) \text{ is } A_1^{3f} \wedge M_4(t-1) \text{ is } A_2^{3f} \wedge M_6(t) \text{ is } A_3^{3f} \wedge M_4(t-2) \text{ is } A_4^{3f} \wedge M_6(t-2) \text{ is } A_5^{3f} \wedge M_8(t) \text{ is } A_6^{3f} \wedge M_8(t-1) \text{ is } A_7^{3f} \wedge M_8(t-4) \text{ is } A_8^{3f} \text{ then } M_4(t) \text{ is } B^{3f}\left(f^{3f}(x), 24.58\right);$$

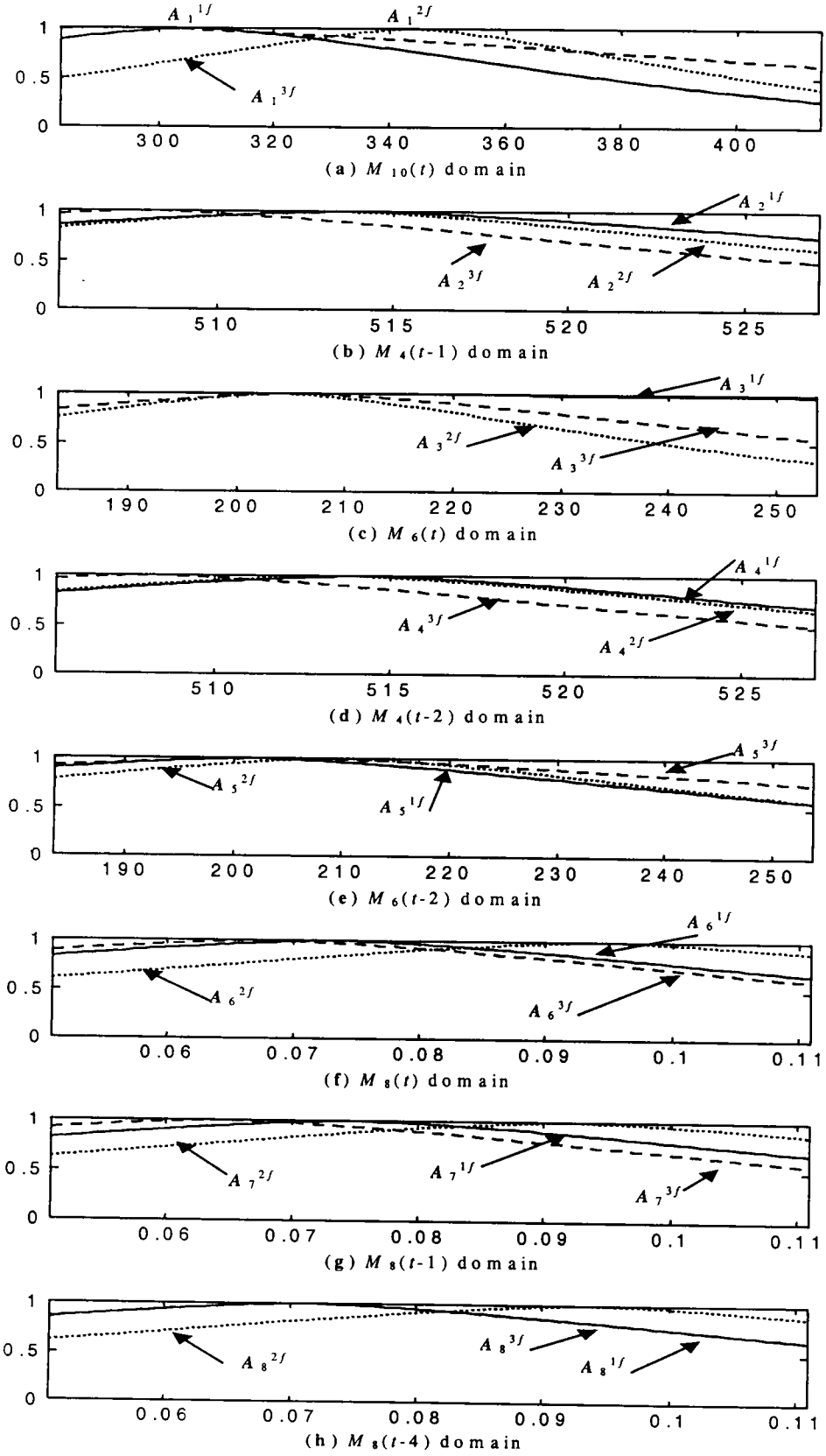


Fig. 6.33: Premise membership functions for class II GFM rules, of $M_4(t)$ variable

where,

$$f^{1f}(x) = 553.45 + 0.12M_{10}(t) - 3.05M_4(t-1) + 0.13M_6(t) + 2.98M_4(t-2) - 0.14M_6(t-2) - 328.41M_8(t) + 405.80M_8(t-1) - 727.65M_8(t-4)$$

$$f^{2f}(x) == -715.76 - 0.20M_{10}(t) + 1.54M_4(t-1) + 0.41M_6(t) + 0.85M_4(t-2) - 0.33M_6(t-2) + 232.37M_8(t) + 22.93M_8(t-1) + 182.37M_8(t-4)$$

$$f^{3f}(x) == -356.06 + 0.11M_{10}(t) + 5.65M_4(t-1) - 0.20M_6(t) - 4.10M_4(t-2) + 0.20M_6(t-2) + 710.64M_8(t) - 467.36M_8(t-1) + 620.78M_8(t-4)$$

The premise variable membership functions A_1^{1f} to A_8^{3f} are shown in Fig.6.33

Since the membership value of A_3^{1f} remains at "1" over the entire domain of the variable $M_6(t)$, this variable can be removed from R^{1f}. The modified form of fuzzy rule can be written as follows:

R^{1f} : if $M_{10}(t)$ is $A_1^{1f} \wedge M_4(t-1)$ is $A_2^{1f} \wedge M_4(t-2)$ is A_4^{1f} ;
$\wedge M_6(t-2)$ is $A_5^{1f} \wedge M_8(t)$ is $A_6^{1f} \wedge M_8(t-1)$ is $A_7^{1f} \wedge M_8(t-4)$ is A_8^{1f} then $M_4(t)$ is $B^{1f}\left(f^{1f}(x), 24.58\right)$

R^{2f} : if $M_{10}(t)$ is $A_1^{2f} \wedge M_4(t-1)$ is $A_2^{2f} \wedge M_6(t)$ is $A_3^{2f} \wedge M_4(t-2)$ is A_4^{2f} ;
$\wedge M_6(t-2)$ is $A_5^{2f} \wedge M_8(t)$ is $A_6^{2f} \wedge M_8(t-1)$ is $A_7^{2f} \wedge M_8(t-4)$ is A_8^{2f} then $M_4(t)$ is $B^{2f}\left(f^{2f}(x), 24.58\right)$

R^{3f} : if $M_{10}(t)$ is $A_1^{3f} \wedge M_4(t-1)$ is $A_2^{3f} \wedge M_6(t)$ is $A_3^{3f} \wedge M_4(t-2)$ is A_4^{3f} ;
$\wedge M_6(t-2)$ is $A_5^{3f} \wedge M_8(t)$ is $A_6^{3f} \wedge M_8(t-1)$ is $A_7^{3f} \wedge M_8(t-4)$ is A_8^{3f} then $M_4(t)$ is $B^{3f}\left(f^{3f}(x), 24.58\right)$

The output of the model and the actual values are plotted in Fig.6.34(a). Figure 6.34(b) shows the model error in per unit.

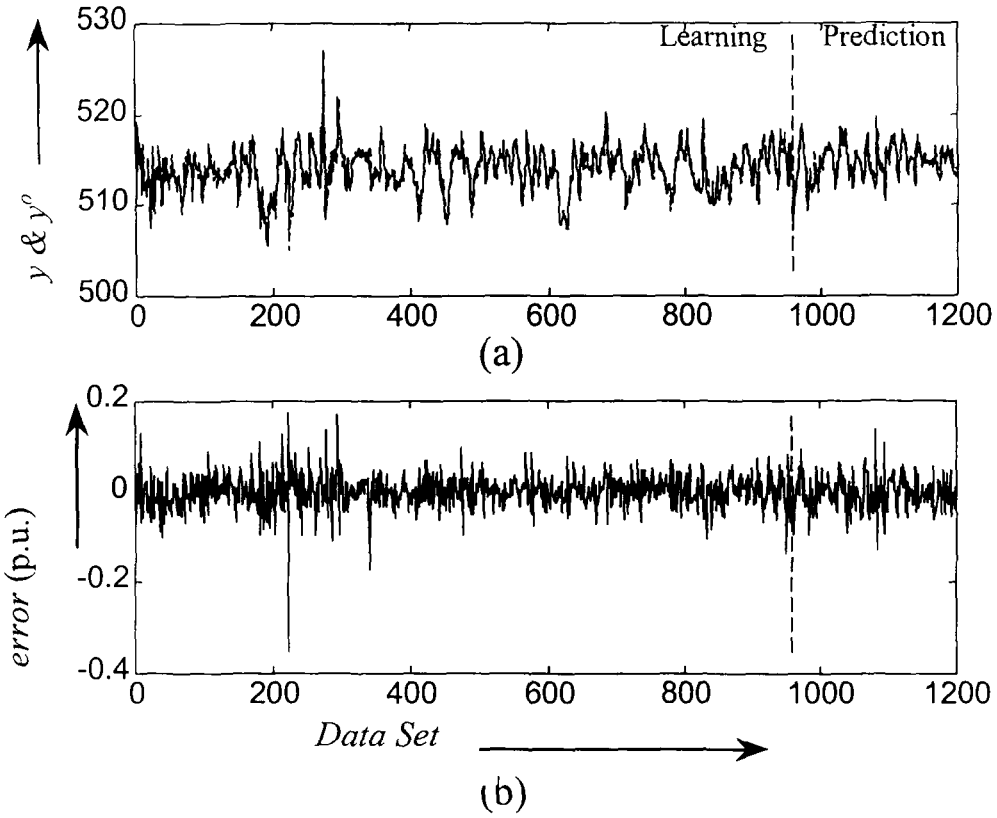


Fig.6.34: Plots of (a) y (————, solid line) and y^o (- - - - - - -,dashed line) (b) *model error* for class II GFM of $M_4(t)$.

Catalyst circulation Rate, $M_5(t)$

This variable is associated with the RRS unit and it depends on $M_{10}(t)$, $M_9(t)$, $M_5(t\text{-}1)$, $M_2(t)$, and $M_3(t\text{-}4)$. The variable $M_5(t)$ exhibits the first order dynamics with the instantaneous effect of Regenerator Bed Temperature, Side Draw off Temperature for HCO, Bottom Vapor Temperature and 40 minutes delayed effect of Flue Gas Temperature. The fuzzy rules are given as follows:

R^{1f} :if $M_{10}(t)$ is $A_1^{1f} \wedge M_9(t)$ is $A_2^{1f} \wedge M_5(t-1)$ is $A_3^{1f} \wedge M_2(t)$ is $A_4^{1f} \wedge M_3(t-4)$ is A_5^{1f} then $M_5(t)$ is $B^{1f}\left(f^{1f}(x), 3.59\right)$;

R^{2f} :if $M_{10}(t)$ is $A_1^{2f} \wedge M_9(t)$ is $A_2^{2f} \wedge M_5(t-1)$ is $A_3^{2f} \wedge M_2(t)$ is $A_4^{2f} \wedge M_3(t-4)$ is A_5^{2f} then $M_5(t)$ is $B^{2f}\left(f^{2f}(x), 3.59\right)$;

R^{3f} :if $M_{10}(t)$ is $A_1^{3f} \wedge M_9(t)$ is $A_2^{3f} \wedge M_5(t-1)$ is $A_3^{3f} \wedge M_2(t)$ is $A_4^{3f} \wedge M_3(t-4)$ is A_5^{3f} then $M_5(t)$ is $B^{3f}\left(f^{3f}(x), 3.59\right)$;

R^{4f} :if $M_{10}(t)$ is $A_1^{4f} \wedge M_9(t)$ is $A_2^{4f} \wedge M_5(t-1)$ is $A_3^{4f} \wedge M_2(t)$ is $A_4^{4f} \wedge M_3(t-4)$ is A_5^{4f} then $M_5(t)$ is $B^{4f}\left(f^{4f}(x), 3.59\right)$;

where,

$$f^{1f}(x) = -43.13 + 0.22M_{10}(t) - 1.32M_9(t) + 4.43M_5(t-1) + 0.09M_2(t) + 0.19M_3(t-4)$$

$$f^{2f}(x) = 86.14 - 0.43M_{10}(t) + 1.39M_9(t) + 11.43M_5(t-1) - 0.40M_2(t) - 0.09M_3(t-4)$$

$$f^{3f}(x) = 25.38 - 0.13M_{10}(t) + 0.40M_9(t) + 0.73M_5(t-1) - 0.06M_2(t) - 0.03M_3(t-4)$$

$$f^{4f}(x) = -18.37 + 0.23M_{10}(t) - 0.11M_9(t) - 13.95M_5(t-1) - 0.25M_2(t) - 0.06M_3(t-4)$$

The premise variable membership functions A_1^{1f} to A_5^{4f} are shown in Fig.6.35

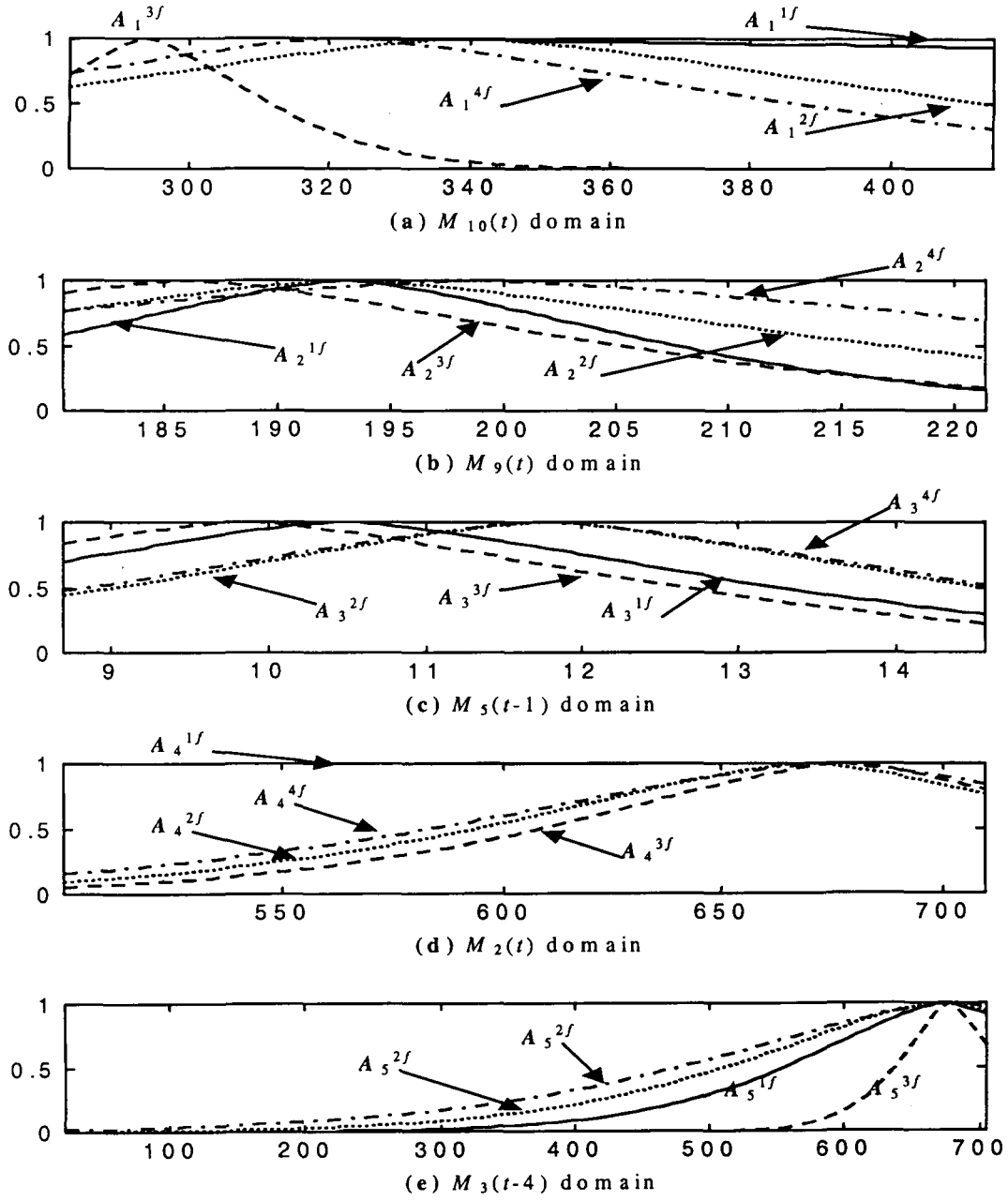


Fig. 6.35: Premise membership functions for class II GFM rules, of $M_5(t)$ variable

Since the membership value of and A_1^{1f} & A_4^{1f} remains at "1" over the entire domain of the variable $M_{10}(t)$ and $M_2(t)$, these variables can be removed R^{1f}. The modified form of fuzzy rule can be written as follows:

R^{1f} : if $M_9(t)$ is $A_2^{1f} \wedge M_5(t-1)$ is $A_3^{1f} \wedge M_3(t-4)$ is A_5^{1f} then $M_5(t)$ is $B^{1f}\left(f^{1f}(x), 3.59\right)$;

R^{2f} : if $M_{10}(t)$ is $A_1^{2f} \wedge M_9(t)$ is $A_2^{2f} \wedge M_5(t-1)$ is $A_3^{2f} \wedge M_2(t)$ is $A_4^{2f} \wedge M_3(t-4)$ is A_5^{2f} then $M_5(t)$ is $B^{2f}\left(f^{2f}(x), 3.59\right)$;

R^{3f} : if $M_{10}(t)$ is $A_1^{3f} \wedge M_9(t)$ is $A_2^{3f} \wedge M_5(t-1)$ is $A_3^{3f} \wedge M_2(t)$ is $A_4^{3f} \wedge M_3(t-4)$ is A_5^{3f} then $M_5(t)$ is $B^{3f}\left(f^{3f}(x), 3.59\right)$;

R^{4f} : if $M_{10}(t)$ is $A_1^{4f} \wedge M_9(t)$ is $A_2^{4f} \wedge M_5(t-1)$ is $A_3^{4f} \wedge M_2(t)$ is $A_4^{4f} \wedge M_3(t-4)$ is A_5^{4f} then $M_5(t)$ is $B^{4f}\left(f^{4f}(x), 3.59\right)$;

The output of the model and the actual values are plotted in Fig.6.36(a). Figure 6.36(b) shows the model error in per unit.

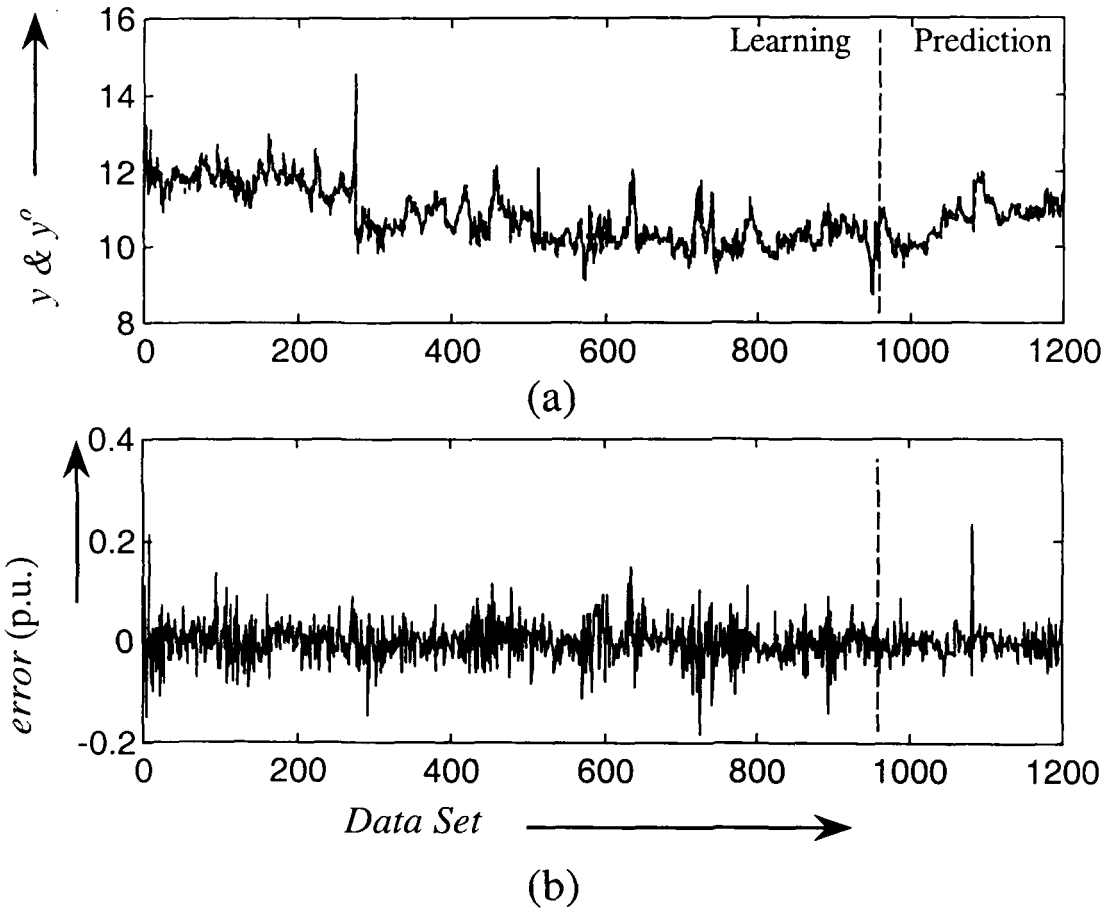


Fig.6.36: Plots of (a) y (————, solid line) and y^o (- - - - - -,dashed line) (b) *model error* for class II GFM of $M_5(t)$.

Compressor Amperage, i.e., $M_6(t)$

This variable is associated with the fractionator unit and it depends on $M_{10}(t)$, $M_6(t\text{-}1)$, and $M_6(t\text{-}2)$. The variable Compressor Amperage, i.e., $M_6(t)$ has the second order dynamics and the instantaneous effect of Bottom Vapor Temperature. The fuzzy rules are given as follows:

$$R^{1f}: \text{if } M_{10}(t) \text{ is } A_1^{1f} \wedge M_6(t-1) \text{ is } A_2^{1f} \wedge M_6(t-2) \text{ is } A_3^{1f} \quad \text{then } M_6(t) \text{ is } B^{1f}\left(f^{1f}(x), 56.13\right);$$

$$R^{2f}: \text{if } M_{10}(t) \text{ is } A_1^{2f} \wedge M_6(t-1) \text{ is } A_2^{2f} \wedge M_6(t-2) \text{ is } A_3^{2f} \quad \text{then } M_6(t) \text{ is } B^{2f}\left(f^{2f}(x), 56.13\right);$$

$$R^{3f}: \text{if } M_{10}(t) \text{ is } A_1^{3f} \wedge M_6(t-1) \text{ is } A_2^{3f} \wedge M_6(t-2) \text{ is } A_3^{3f} \quad \text{then } M_6(t) \text{ is } B^{3f}\left(f^{3f}(x), 56.13\right);$$

where,

$$f^{1f}(x) = -328.80 + \quad 1.46M_6(t-1) + 1.23M_6(t-2)$$

$$f^{2f}(x) = -780.80 + 0.95M_{10}(t) - 0.29M_6(t-1) + 3.07M_6(t-2)$$

$$f^{3f}(x) = 104.88 + 0.78M_{10}(t) + 2.03M_6(t-1) - 2.22M_6(t-2)$$

The premise variable membership functions A_1^{1f} to A_3^{3f} are shown in Fig.6.37

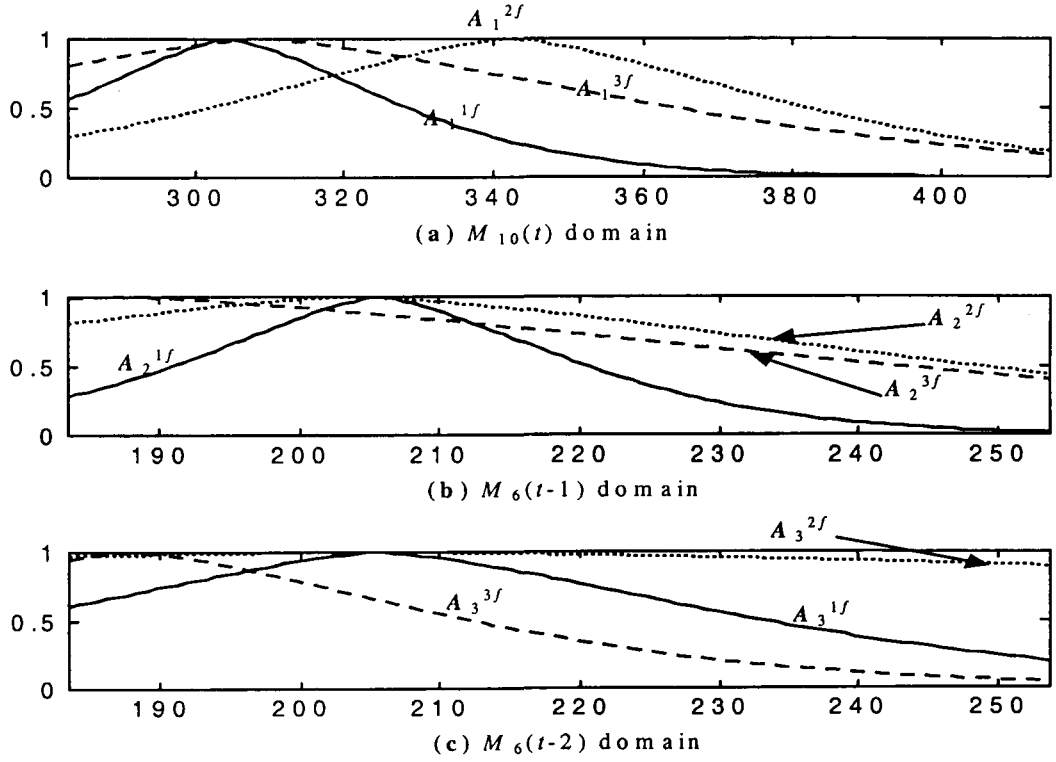


Fig. 6.37: Premise membership functions for class II GFM rules, of $M_6(t)$ variable

Since the membership value of A_3^{2f} almost remains nearly "1" over the entire domain of the variable $M_6(t\text{-}2)$, the variable $M_6(t\text{-}2)$ can be removed from the R^{2f}. The modified form of fuzzy rule can be written as follows:

$$R^{1f}: \text{if } M_{10}(t) \text{ is } A_1^{1f} \wedge M_6(t-1) \text{ is } A_2^{1f} \wedge M_6(t-2) \text{ is } A_3^{1f} \quad \text{then } M_6(t) \text{ is } B^{1f}\left(f^{1f}(x), 56.13\right);$$

$$R^{2f}: \text{if } M_{10}(t) \text{ is } A_1^{2f} \wedge M_6(t-1) \text{ is } A_2^{2f} \quad \text{then } M_6(t) \text{ is } B^{2f}\left(f^{2f}(x), 56.13\right);$$

$$R^{3f}: \text{if } M_{10}(t) \text{ is } A_1^{3f} \wedge M_6(t-1) \text{ is } A_2^{3f} \wedge M_6(t-2) \text{ is } A_3^{3f} \quad \text{then } M_6(t) \text{ is } B^{3f}\left(f^{3f}(x), 56.13\right);$$

The output of the model and the actual values are plotted in Fig.6.38(a). Figure 6.38(b) shows the model error in per unit.

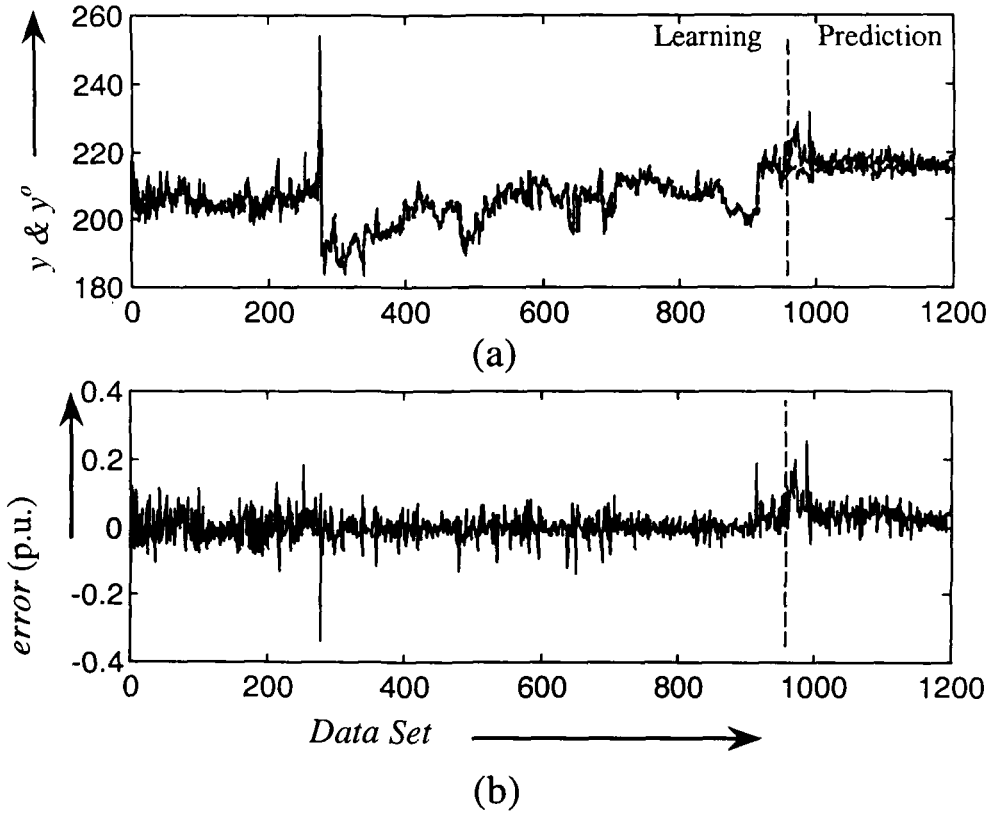


Fig.6.38: Plots of (a) y (————, solid line) and y^o (- - - - - -,dashed line) (b) *model error* for class II GFM of $M_6(t)$.

Fractionator Top Temperature., $M_7(t)$

This variable is associated with the fractionator unit and it depends on $M_{10}(t\text{-}2)$, M_9 $(t\text{-}2)$, M_6 $(t\text{-}2)$, M_9 (t), $M_9(t\text{-}4)$, $M_9(t\text{-}3)$, and $M_7(t\text{-}1)$. The variable $M_7(t)$ displays the first order dynamics with at least the instantaneous effect of Side Draw off Temperature for HCO, and 20 minutes delayed effect of Compressor Amperage as well as Bottom Vapor Temperature. The fuzzy rules are given as follows:

$$R^{1f}: \text{if } M_{10}(t-2) \text{ is } A_1^{1f} \wedge M_9(t-2) \text{ is } A_2^{1f} \wedge M_6(t-2) \text{ is } A_3^{1f} \wedge M_9(t) \text{ is } A_4^{1f} \wedge M_9(t-4) \text{ is } A_5^{1f} \wedge M_9(t-3) \text{ is } A_6^{1f} \wedge M_7(t-1) \text{ is } A_7^{1f} \quad \text{then } M_7(t) \text{ is } B^{1f}\left(f^{1f}(x), 34.65\right);$$

$$R^{2f}: \text{if } M_{10}(t-2) \text{ is } A_1^{2f} \wedge M_9(t-2) \text{ is } A_2^{2f} \wedge M_6(t-2) \text{ is } A_3^{2f} \wedge M_9(t) \text{ is } A_4^{2f} \wedge M_9(t-4) \text{ is } A_5^{2f} \wedge M_9(t-3) \text{ is } A_6^{2f} \wedge M_7(t-1) \text{ is } A_7^{2f} \quad \text{then } M_7(t) \text{ is } B^{2f}\left(f^{2f}(x), 34.65\right);$$

$$R^{3f}: \text{if } M_{10}(t-2) \text{ is } A_1^{3f} \wedge M_9(t-2) \text{ is } A_2^{3f} \wedge M_6(t-2) \text{ is } A_3^{3f} \wedge M_9(t) \text{ is } A_4^{3f} \wedge M_9(t-4) \text{ is } A_5^{3f} \wedge M_9(t-3) \text{ is } A_6^{3f} \wedge M_7(t-1) \text{ is } A_7^{3f} \quad \text{then } M_7(t) \text{ is } B^{3f}\left(f^{3f}(x), 34.65\right);$$

where,

$$f^{1f}(x) = 17.43 - 0.27M_9(t-2) - 0.01M_6(t-2) + 0.30M_9(t) - 0.05M_9(t-4) + 0.08M_9(t-3) + 0.77M_7(t-1)$$

$$f^{2f}(x) = 32.42 - 0.08M_9(t-2) - 0.02M_6(t-2) + 0.65M_9(t) - 0.04M_9(t-4) - 0.17M_9(t-3) + 0.07M_7(t-1)$$

$$f^{3f}(x) = -577.56 + 0.08M_{10}(t-2) - 0.14M_9(t-2) - 0.01M_6(t-2) + 3.50M_9(t) + 0.77M_9(t-4) - 0.36M_9(t-3) - 0.34M_7(t-1)$$

The premise variable membership functions A_1^{1f} to A_7^{3f} are shown in Fig.6.39.

Since the membership value of A_1^{1f}, A_1^{3f} remains nearly "1" almost over the entire domain of the variable $M_{10}(t\text{-}2)$, the variable $M_{10}(t\text{-}2)$ can be removed from the R^{1f} and R^{3f}. Similarly $M_{10}(t\text{-}2)$ can be removed from the R^{1f}, because of A_3^{1f} value over its entire domain and $M_9(t)$ can be removed from the R^{1f} and R^{3f} because of values for A_4^{1f} and A_4^{3f}. $M_9(t\text{-}4)$ is removed from the R^{1f}, R^{2f} and R^{3f} because of values for A_5^{1f}, A_5^{2f} and A_5^{3f}.

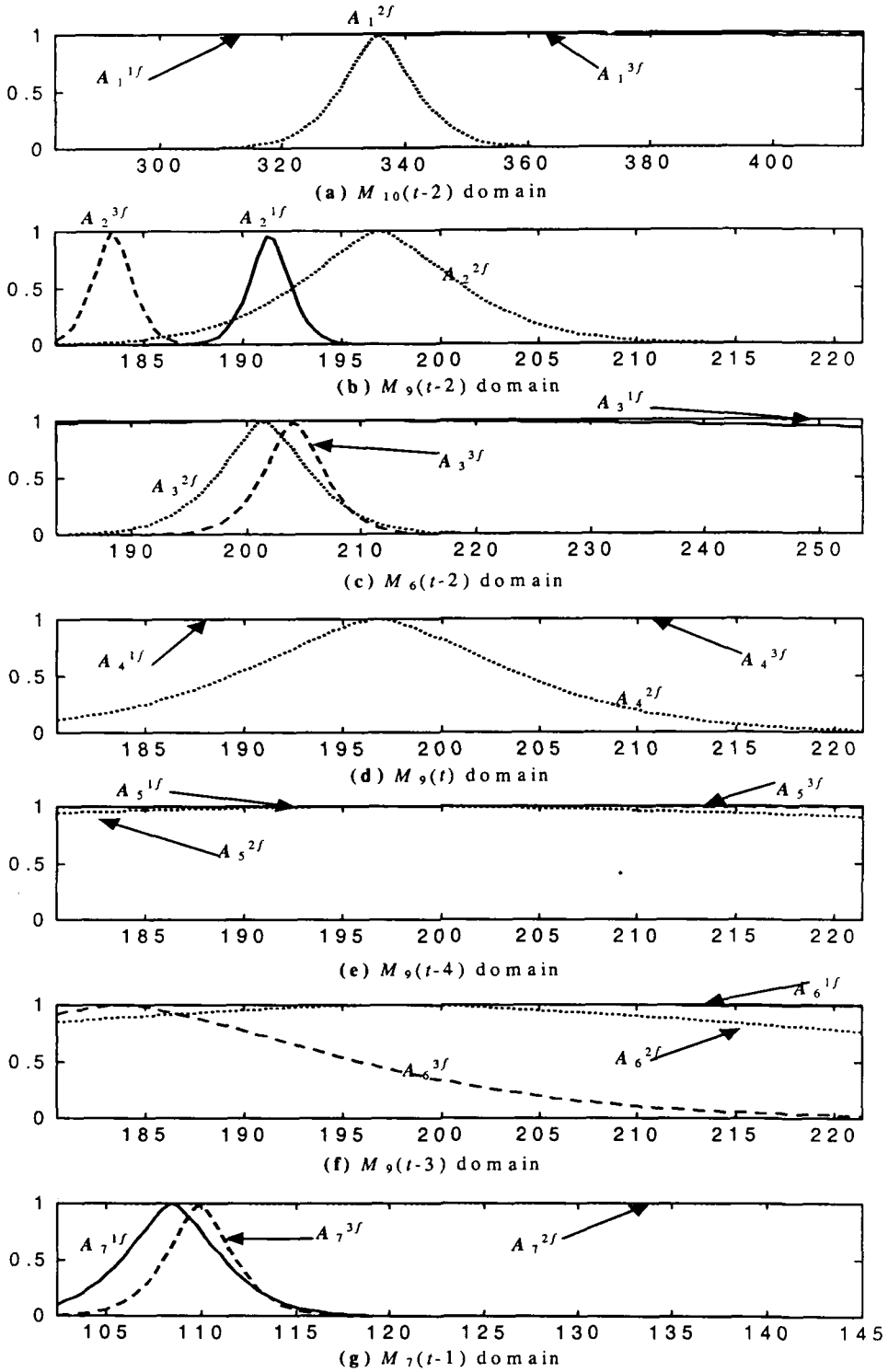


Fig. 6.39: Premise membership functions for class II GFM rules, of $M_7(t)$ variable.

Moreover, $M_9(t\text{-}3)$ and $M_7(t\text{-}1)$ is removed from the R^{1f} and R^{2f}, respectively, because of values for A_6^{1f} and A_7^{2f}. The modified form of fuzzy rule can be written as follows:

$$R^{1f}: \text{if } M_9(t-2) \text{ is } A_2^{1f} \wedge M_7(t-1) \text{ is } A_7^{1f} \text{ then } M_7(t) \text{ is } B^{1f}\left(f^{1f}(x),\ 34.65\right);$$

$$R^{2f}: \text{if } M_{10}(t-2) \text{ is } A_1^{2f} \wedge M_9(t-2) \text{ is } A_2^{2f} \wedge M_6(t-2) \text{ is } A_3^{2f} \wedge M_9(t) \text{ is } A_4^{2f} \wedge M_9(t-3) \text{ is } A_6^{2f} \text{ then } M_7(t) \text{ is } B^{2f}\left(f^{2f}(x),\ 34.65\right);$$

$$R^{3f}: \text{if } M_9(t-2) \text{ is } A_2^{3f} \wedge M_6(t-2) \text{ is } A_3^{3f} \wedge M_9(t-3) \text{ is } A_6^{3f} \wedge M_7(t-1) \text{ is } A_7^{3f} \text{ then } M_7(t) \text{ is } B^{3f}\left(f^{3f}(x),\ 34.65\right);$$

The output of the model and the actual values are plotted in Fig.6.40(a). Figure 6.40(b) shows the model error in per unit.

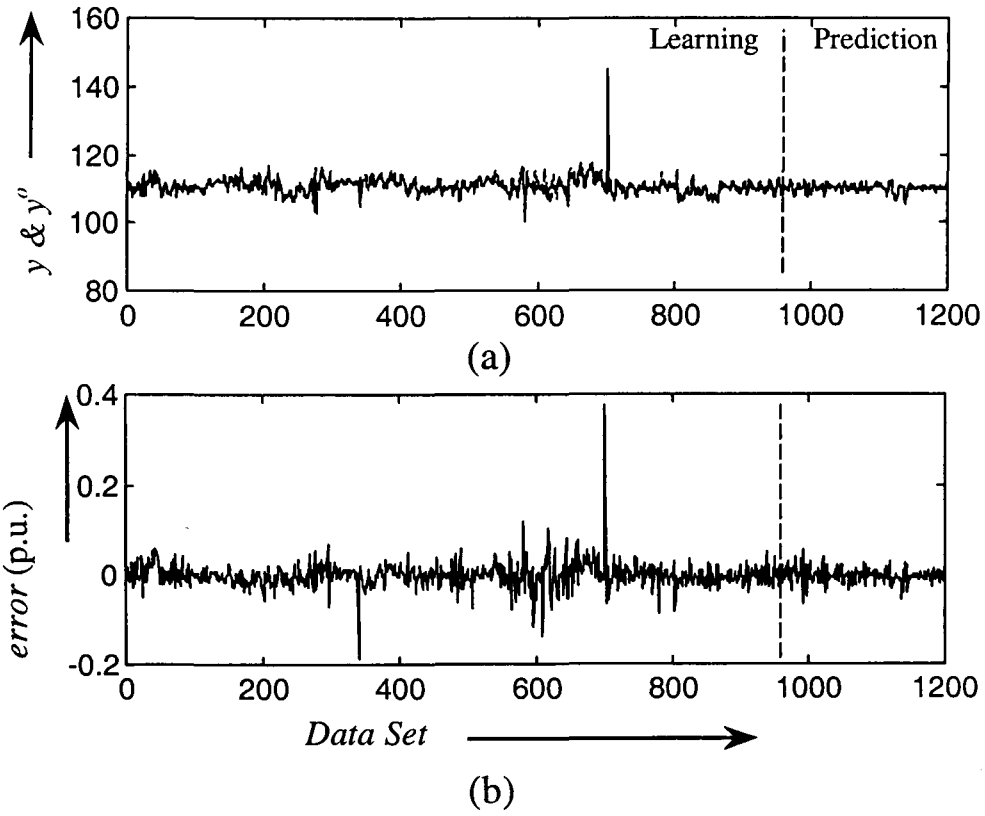


Fig.6.40: Plots of (a) y (_____, solid line) and y^o (- - - - - -,dashed line) (b) *model error* for class II GFM of $M_7(t)$.

Delta* P *across the Fractionator, $M_8(t)$

This variable is associated with the fractionator unit and it depends on $M_{10}(t)$, $M_{10}(t\text{-}2)$, $M_9(t)$, $M_8(t\text{-}1)$, $M_6(t)$, $M_9(t\text{-}2)$, and $M_5(t\text{-}2)$. This variable exhibits the first order dynamics with at least the instantaneous effect of Compressor Amperage, Side Draw off Temperature for HCO, Bottom Vapor Temperature and 20 minutes delayed effect of Compressor Amperage. The fuzzy rules are given as follows:

$$R^{1f}: \text{if } M_{10}(t) \text{ is } A_1^{1f} \wedge M_{10}(t-2) \text{is } A_2^{1f} \wedge M_9(t) \text{ is } A_3^{1f} \wedge M_8(t-1) \text{ is } A_4^{1f} \wedge M_6(t) \text{ is } A_5^{1f} \wedge M_9(t-2) \text{ is } A_6^{1f} \wedge M_5(t-2) \text{ is } A_7^{1f} \quad \text{then } M_8(t) \text{ is } B^{1f}\left(f^{1f}(x), 0.13\right);$$

$$R^{2f}: \text{if } M_{10}(t) \text{ is } A_1^{2f} \wedge M_{10}(t-2) \text{is } A_2^{2f} \wedge M_9(t) \text{ is } A_3^{2f} \wedge M_8(t-1) \text{ is } A_4^{2f} \wedge M_6(t) \text{ is } A_5^{2f} \wedge M_9(t-2) \text{ is } A_6^{2f} \wedge M_5(t-2) \text{ is } A_7^{2f} \quad \text{then } M_8(t) \text{ is } B^{2f}\left(f^{2f}(x), 0.08\right);$$

$$R^{3f}: \text{if } M_{10}(t) \text{ is } A_1^{3f} \wedge M_{10}(t-2) \text{is } A_2^{3f} \wedge M_9(t) \text{ is } A_3^{3f} \wedge M_8(t-1) \text{ is } A_4^{3f} \wedge M_6(t) \text{ is } A_5^{3f} \wedge M_9(t-2) \text{ is } A_6^{3f} \wedge M_5(t-2) \text{ is } A_7^{3f} \quad \text{then } M_8(t) \text{ is } B^{3f}\left(f^{3f}(x), 0.14\right);$$

$$R^{4f}: \text{if } M_{10}(t) \text{ is } A_1^{4f} \wedge M_{10}(t-2) \text{is } A_2^{4f} \wedge M_9(t) \text{ is } A_3^{4f} \wedge M_8(t-1) \text{ is } A_4^{4f} \wedge M_6(t) \text{ is } A_5^{4f} \wedge M_9(t-2) \text{ is } A_6^{4f} \wedge M_5(t-2) \text{ is } A_7^{4f} \quad \text{then } M_8(t) \text{ is } B^{4f}\left(f^{4f}(x), 0.08\right);$$

$$R^{5f}: \text{if } M_{10}(t) \text{ is } A_1^{5f} \wedge M_{10}(t-2) \text{is } A_2^{5f} \wedge M_9(t) \text{ is } A_3^{5f} \wedge M_8(t-1) \text{ is } A_4^{5f} \wedge M_6(t) \text{ is } A_5^{5f} \wedge M_9(t-2) \text{ is } A_6^{5f} \wedge M_5(t-2) \text{ is } A_7^{5f} \quad \text{then } M_8(t) \text{ is } B^{5f}\left(f^{5f}(x), 0.16\right);$$

where,

$$f^{1f}(x) = -0.55 - 0.72M_8(t-1)$$

$$f^{2f}(x) = -10.51 + 0.01M_{10}(t) + 0.02M_{10}(t-2) + 0.01M_9(t) + 0.77M_8(t-1) + 0.02M_6(t) - 0.01M_9(t-2) - 0.15M_5(t-2)$$

$$f^{3f}(x) = -0.02 + 1.05M_8(t-1)$$

$$f^{4f}(x) = -2.98 + 0.01M_9(t) - 2.70M_8(t-1) - 0.04M_5(t-2)$$

$$f^{5f}(x) = -0.03 + 0.38M_8(t-1)$$

The premise variable membership functions A_1^{1f} to A_7^{3f} are shown in Fig.6.41. Since the membership value of A_5^{1f} and A_6^{5f} remains nearly "1" almost over the entire domain of the variable $M_6(t)$ and $M_9(t\text{-}2)$,respectively, the variable $M_6(t)$ can be removed from the R^{1f} and $M_9(t\text{-}2)$ from R^{3f}. Moreover, A_4^{2f} remains nearly to "0" over the entire domain of $M_8(t\text{-}1)$, rule R^{2f} is removed from the fuzzy model. After removing the

rule R^{2f} model is again relearn but neither the model updated nor its performance gets changed.

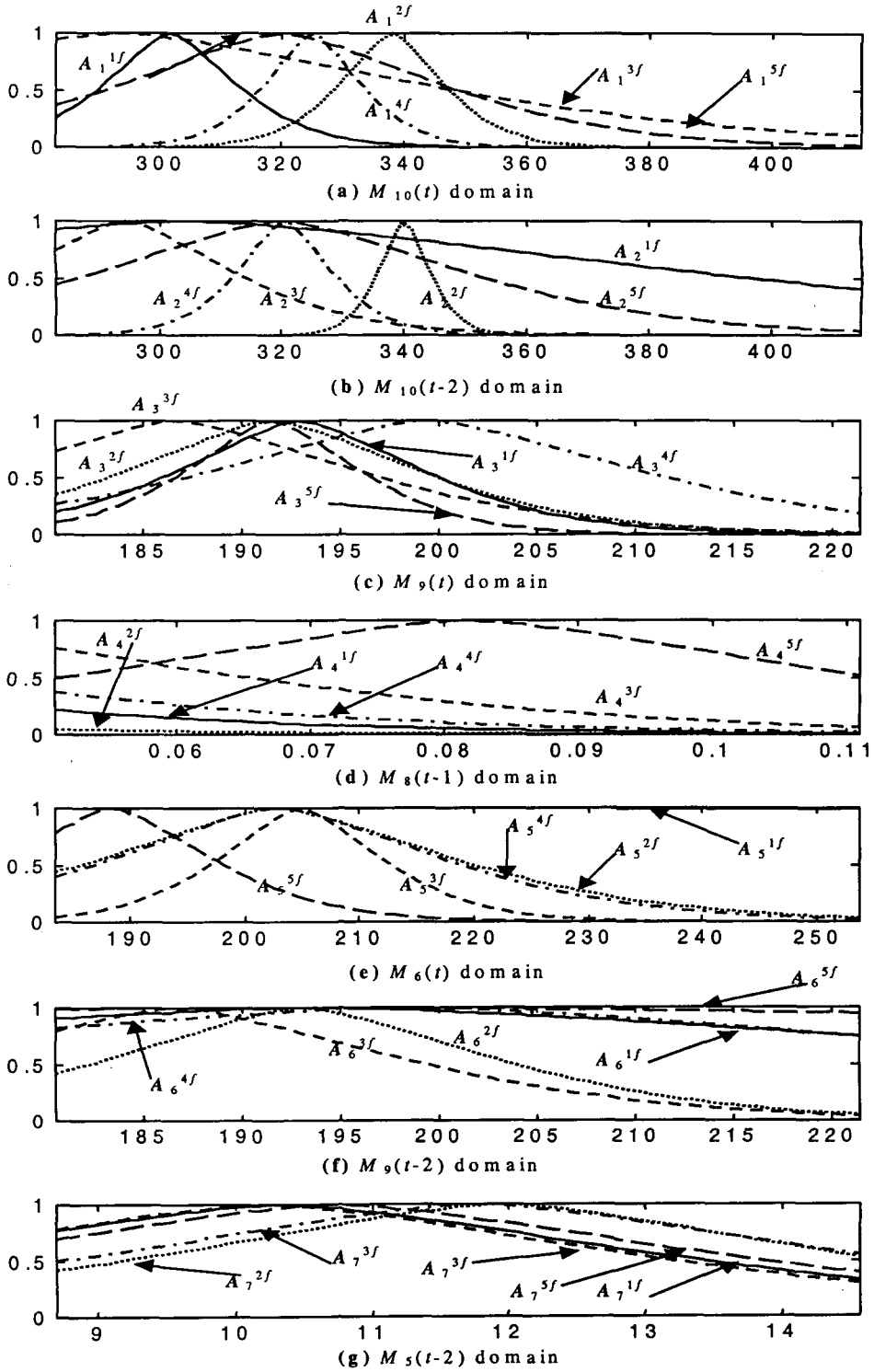


Fig. 6.41: Premise membership functions for class II GFM rules, of $M_8(t)$ variable

The modified form of fuzzy rule can be written as follows:

$$R^{1f}: \text{if } M_{10}(t) \text{ is } A_1^{1f} \wedge M_{10}(t-2) \text{is } A_2^{1f} \wedge M_9(t) \text{ is } A_3^{1f} \wedge M_8(t-1) \text{ is } A_4^{1f} \wedge M_9(t-2) \text{ is } A_6^{1f} \wedge M_5(t-2) \text{ is } A_7^{1f} \quad \text{then } M_8(t) \text{ is } B^{1f}\left(f^{1f}(x), 0.13\right);$$

$$R^{3f}: \text{if } M_{10}(t) \text{ is } A_1^{3f} \wedge M_{10}(t-2) \text{is } A_2^{3f} \wedge M_9(t) \text{ is } A_3^{3f} \wedge M_8(t-1) \text{ is } A_4^{3f} \wedge M_6(t) \text{ is } A_5^{3f} \wedge M_9(t-2) \text{ is } A_6^{3f} \wedge M_5(t-2) \text{ is } A_7^{3f} \quad \text{then } M_8(t) \text{ is } B^{3f}\left(f^{3f}(x), 0.14\right);$$

$$R^{4f}: \text{if } M_{10}(t) \text{ is } A_1^{4f} \wedge M_{10}(t-2) \text{is } A_2^{4f} \wedge M_9(t) \text{ is } A_3^{4f} \wedge M_8(t-1) \text{ is } A_4^{4f} \wedge M_6(t) \text{ is } A_5^{4f} \wedge M_9(t-2) \text{ is } A_6^{4f} \wedge M_5(t-2) \text{ is } A_7^{4f} \quad \text{then } M_8(t) \text{ is } B^{4f}\left(f^{4f}(x), 0.08\right);$$

$$R^{5f}: \text{if } M_{10}(t) \text{ is } A_1^{5f} \wedge M_{10}(t-2) \text{is } A_2^{5f} \wedge M_9(t) \text{ is } A_3^{5f} \wedge M_8(t-1) \text{ is } A_4^{5f} \wedge M_6(t) \text{ is } A_5^{5f} \wedge M_5(t-2) \text{ is } A_7^{5f} \quad \text{then } M_8(t) \text{ is } B^{5f}\left(f^{5f}(x), 0.16\right);$$

The output of the model and the actual values are plotted in Fig.6.42(a). Figure 6.42(b) shows the model error in per unit.

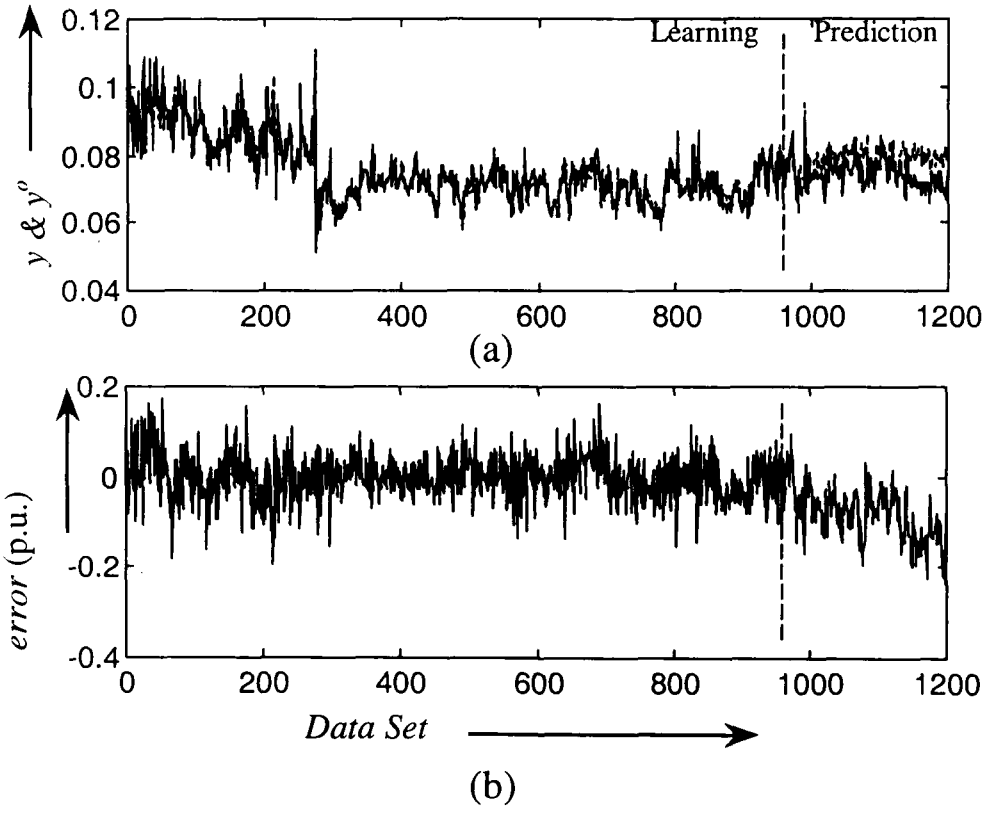


Fig.6.42: Plots of (a) y (————, solid line) and y^o (- - - - - -,dashed line) (b) *model error* for class II GFM of $M_8(t)$.

Side Draw off Temperature for HCO, $M_9(t)$

This variable is associated with the fractionator unit and it depends on M_{10} (t), $M_9(t\text{-}4)$, $M_9(t\text{-}3)$, M_{10} $(t\text{-}3)$, $M_9(t\text{-}2)$, and $M_9(t\text{-}1)$. The model of the variable $M_9(t)$ has the fourth order dynamics with at least the instantaneous effect of Bottom Vapor Temperature. The fuzzy rules are given as follows:

$$R^{1f}: \text{if } M_{10}(t) \text{ is } A_1^{1f} \wedge M_9(t-4) \text{ is } A_2^{1f} \wedge M_9(t-3) \text{ is } A_3^{1f} \wedge M_{10}(t-3) \text{ is } A_4^{1f} \wedge M_9(t-2) \text{ is } A_5^{1f} \wedge M_9(t-1) \text{ is } A_6^{1f} \quad \text{then } M_9(t) \text{ is } B^{1f}\left(f^{1f}(x), 40.91\right);$$

$$R^{2f}: \text{if } M_{10}(t) \text{ is } A_1^{2f} \wedge M_9(t-4) \text{ is } A_2^{2f} \wedge M_9(t-3) \text{ is } A_3^{2f} \wedge M_{10}(t-3) \text{ is } A_4^{2f} \wedge M_9(t-2) \text{ is } A_5^{2f} \wedge M_9(t-1) \text{ is } A_6^{2f} \quad \text{then } M_9(t) \text{ is } B^{2f}\left(f^{2f}(x), 40.91\right);$$

$$R^{3f}: \text{if } M_{10}(t) \text{ is } A_1^{3f} \wedge M_9(t-4) \text{ is } A_2^{3f} \wedge M_9(t-3) \text{ is } A_3^{3f} \wedge M_{10}(t-3) \text{ is } A_4^{3f} \wedge M_9(t-2) \text{ is } A_5^{3f} \wedge M_9(t-1) \text{ is } A_6^{3f} \quad \text{then } M_9(t) \text{ is } B^{3f}\left(f^{3f}(x), 40.91\right);$$

where,

$$f^{1f}(x) = -212.13 - 0.06M_{10}(t) + 0.57M_9(t-4) - 0.29M_9(t-3) + 0.37M_{10}(t-3) - 1.24M_9(t-2) + 2.49M_9(t-1)$$

$$f^{2f}(x) = -41.81 + 0.03M_{10}(t) + 0.61M_9(t-4) - 0.43M_9(t-3) + 0.12M_{10}(t-3) - 0.37M_9(t-2) + 1.13M_9(t-1)$$

$$f^{3f}(x) = 95.72 + 0.26M_{10}(t) - 0.28M_9(t-4) + 0.34M_9(t-3) - 0.43M_{10}(t-3) + 1.51M_9(t-2) - 0.74M_9(t-1)$$

The premise variable membership functions A_1^{1f} to A_6^{3f} are shown in Fig.6.43.
Since the membership value of A_3^{1f}, A_3^{3f} and A_3^{3f} remains nearly "1" almost over the entire domain of the variable $M_9(t\text{-}3)$, the variable $M_9(t\text{-}3)$, can be removed from the R^{1f} R^{2f} and R^{3f}. Similarly $M_9(t\text{-}4)$ can be removed from the R^{1f}, because of A_2^{1f} value over its entire domain, $M_9(t\text{-}2)$ is removed from the R^{1f} because of values for A_5^{1f} and $M_{10}(t\text{-}3)$ is removed from the R^{2f} because of values for A_4^{2f}.

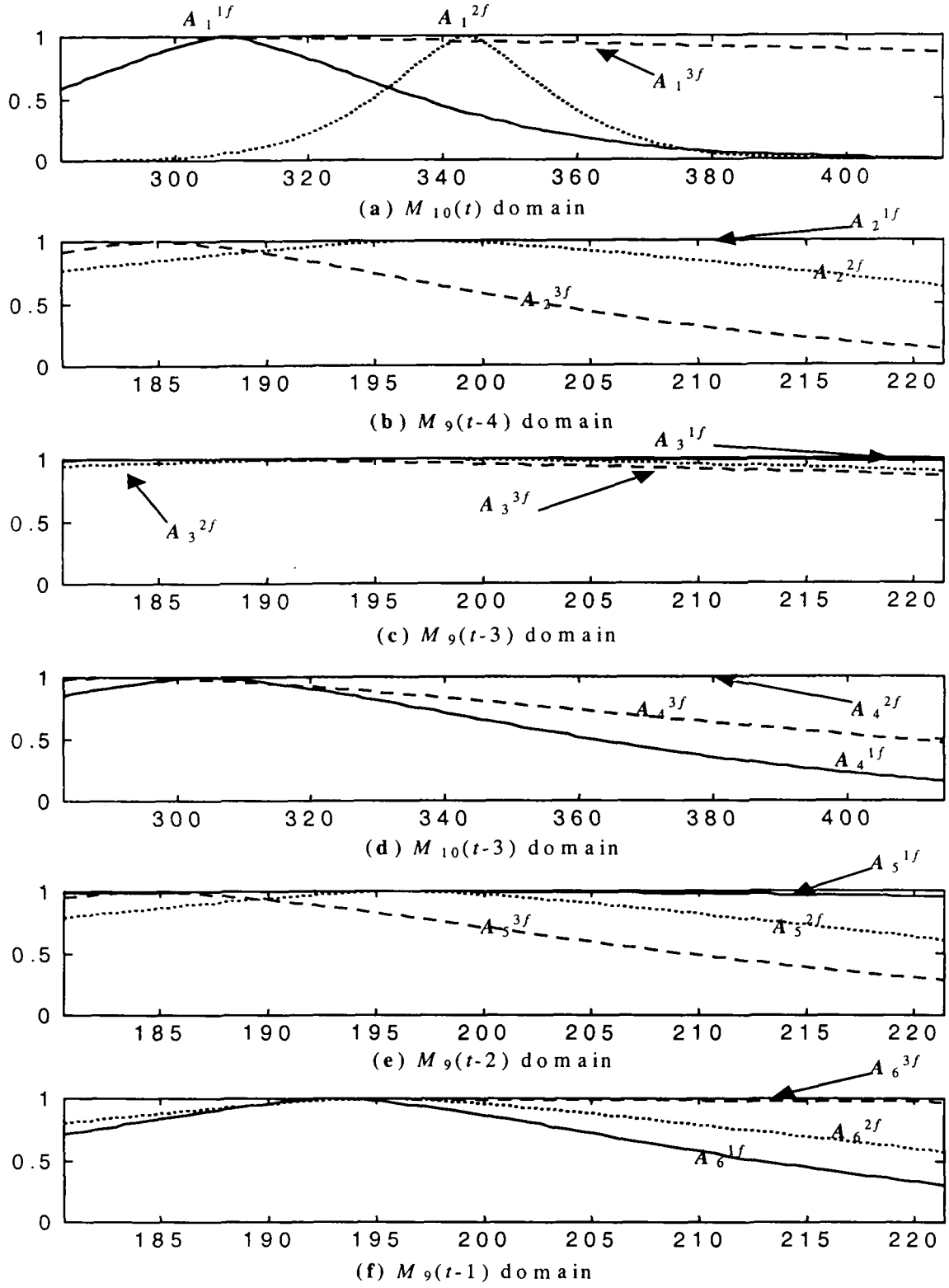


Fig. 6.43: Premise membership functions for class II GFM rules, of $M_9(t)$ variable

Moreover, $M_{10}(t)$ and $M_9(t\text{-}1)$ is removed from the R^{3f}, because of values for A_1^{3f} and A_6^{3f}. The modified form of fuzzy rule can be written as follows:

R^{1f} : if $M_{10}(t)$ is $A_1^{1f} \wedge M_{10}(t-3)$ is $A_4^{1f} \wedge M_9(t-1)$ is A_6^{1f} then $M_9(t)$ is $B^{1f}\left(f^{1f}(x), 40.91\right)$;

R^{2f} : if $M_{10}(t)$ is $A_1^{2f} \wedge M_9(t-4)$ is A_2^{2f}
$\wedge M_9(t-2)$ is $A_5^{2f} \wedge M_9(t-1)$ is A_6^{2f} then $M_9(t)$ is $B^{2f}\left(f^{2f}(x), 40.91\right)$;

R^{3f} : if $M_9(t-4)$ is $A_2^{3f} \wedge M_{10}(t-3)$ is $A_4^{3f} \wedge M_9(t-2)$ is A_5^{3f} then $M_9(t)$ is $B^{3f}\left(f^{3f}(x), 40.91\right)$;

The output of the model and the actual values are plotted in Fig.6.44 (a). Figure 6.44(b) shows the model error in per unit.

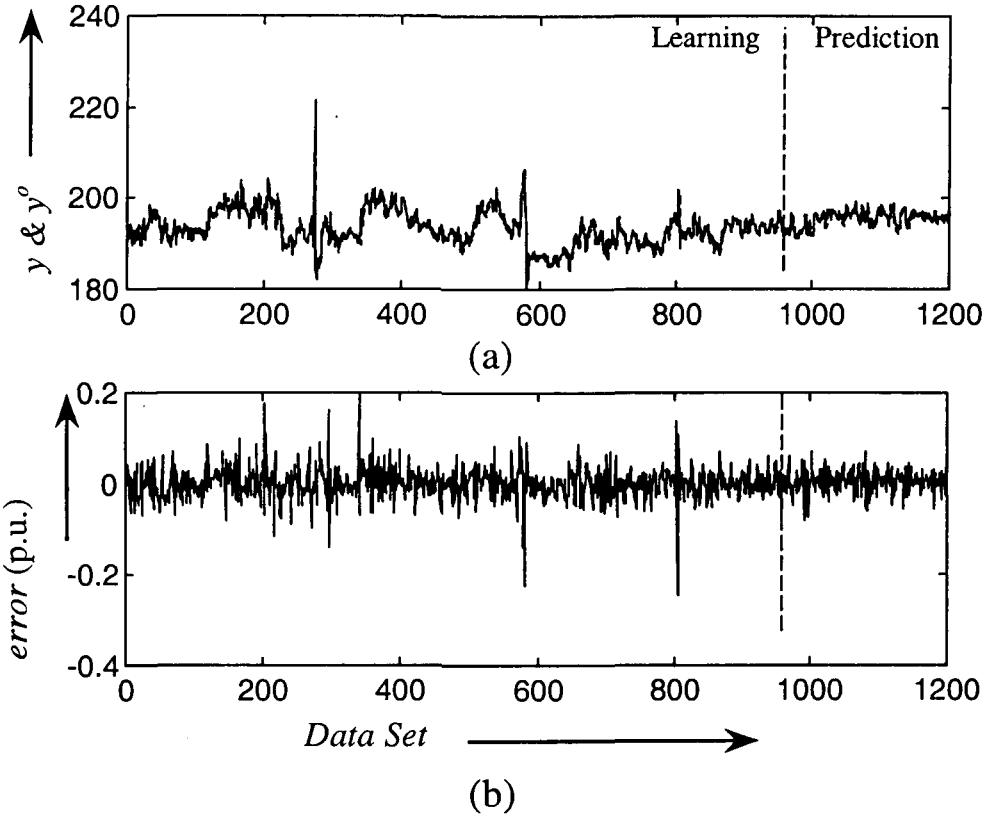


Fig.6.44: Plots of (a) y (————, solid line) and y^o (- - - - - -,dashed line) (b) *model error* for class II GFM of $M_9(t)$.

Bottom Vapor Temperature, $M_{10}(t)$

This variable is associated with the fractionator unit and it depends on $M_9(t)$, $M_5(t)$, $M_8(t)$, $M_6(t)$, $M_{10}(t\text{-}1)$, $M_{10}(t\text{-}3)$, and $M_8(t\text{-}1)$. The model for the variable $M_{10}(t)$

has the third order dynamics with the instantaneous effect of Catalyst Circulation Rate, Compressor Amperage, Delta P across the Fractionator and Side Draw off Temperature for HCO. The fuzzy rules are given as follows:

$$R^{1f}: \text{if } M_9(t) \text{ is } A_1^{1f} \wedge M_5(t) \text{ is } A_2^{1f} \wedge M_8(t) \text{ is } A_3^{1f} \wedge M_6(t) \text{ is } A_4^{1f} \wedge M_{10}(t-1) \text{ is } A_5^{1f} \wedge M_{10}(t-3) \text{ is } A_6^{1f} \wedge M_8(t-1) \text{ is } A_7^{1f} \text{ then } M_{10}(t) \text{ is } B^{1f}\left(f^{1f}(x), 125.39\right);$$

$$R^{2f}: \text{if } M_9(t) \text{ is } A_1^{2f} \wedge M_5(t) \text{ is } A_2^{2f} \wedge M_8(t) \text{ is } A_3^{2f} \wedge M_6(t) \text{ is } A_4^{2f} \wedge M_{10}(t-1) \text{ is } A_5^{2f} \wedge M_{10}(t-3) \text{ is } A_6^{2f} \wedge M_8(t-1) \text{ is } A_7^{2f} \text{ then } M_{10}(t) \text{ is } B^{2f}\left(f^{2f}(x), 125.39\right);$$

$$R^{3f}: \text{if } M_9(t) \text{ is } A_1^{3f} \wedge M_5(t) \text{ is } A_2^{3f} \wedge M_8(t) \text{ is } A_3^{3f} \wedge M_6(t) \text{ is } A_4^{3f} \wedge M_{10}(t-1) \text{ is } A_5^{3f} \wedge M_{10}(t-3) \text{ is } A_6^{3f} \wedge M_8(t-1) \text{ is } A_7^{3f} \text{ then } M_{10}(t) \text{ is } B^{3f}\left(f^{3f}(x), 125.39\right);$$

where,

$$f^{1f}(x) = -1149.90 + 1.01M_9(t) - 8.85M_5(t) - 2038.80M_8(t) + 3.80M_6(t) + 1.47M_{10}(t-1) + 1.06M_{10}(t-3) - 1477.10M_8(t-1)$$

$$f^{2f}(x) = -1041.00 + 1.37M_9(t) + 1.19M_5(t) - 1091.40M_8(t) + 3.33M_6(t) + 0.90M_{10}(t-1) + 0.46M_{10}(t-3) - 204.70M_8(t-1)$$

$$f^{3f}(x) = 1129.50 - 0.93M_9(t) + 16.43M_5(t) - 2849.10M_8(t) - 4.24M_6(t) + 0.55M_{10}(t-1) - 1.02M_{10}(t-3) + 853.90M_8(t-1)$$

The premise variable membership functions A_1^{1f} to A_7^{3f} are shown in Fig.6.45.

Since the membership value of A_6^{6f} and A_6^{6f} remains nearly "1" almost over the entire domain of the variable $M_{10}(t\text{-}3)$, the variable $M_{10}(t\text{-}3)$, can be removed from R^{2f} and R^{3f}. Similarly $M_9(t)$ can be removed from the R^{1f}, because of A_1^{1f} value over its entire domain, and $M_6(t)$ is removed from the R^{2f} because of values for A_4^{2f}. The modified form of fuzzy rule can be written as follows:

$$R^{1f}: \text{if } M_5(t) \text{ is } A_2^{1f} \wedge M_8(t) \text{ is } A_3^{1f} \wedge M_6(t) \text{ is } A_4^{1f} \wedge M_{10}(t-1) \text{ is } A_5^{1f} \wedge M_{10}(t-3) \text{ is } A_6^{1f} \wedge M_8(t-1) \text{ is } A_7^{1f} \text{ then } M_{10}(t) \text{ is } B^{1f}\left(f^{1f}(x), 125.39\right);$$

$$R^{2f}: \text{if } M_9(t) \text{ is } A_1^{2f} \wedge M_5(t) \text{ is } A_2^{2f} \wedge M_8(t) \text{ is } A_3^{2f} \wedge M_{10}(t-1) \text{ is } A_5^{2f} \wedge M_8(t-1) \text{ is } A_7^{2f} \text{ then } M_{10}(t) \text{ is } B^{2f}\left(f^{2f}(x), 125.39\right);$$

$$R^{3f}: \text{if } M_9(t) \text{ is } A_1^{3f} \wedge M_5(t) \text{ is } A_2^{3f} \wedge M_8(t) \text{ is } A_3^{3f} \wedge M_6(t) \text{ is } A_4^{3f} \wedge M_{10}(t-1) \text{ is } A_5^{3f} \wedge M_8(t-1) \text{ is } A_7^{3f} \text{ then } M_{10}(t) \text{ is } B^{3f}\left(f^{3f}(x), 125.39\right);$$

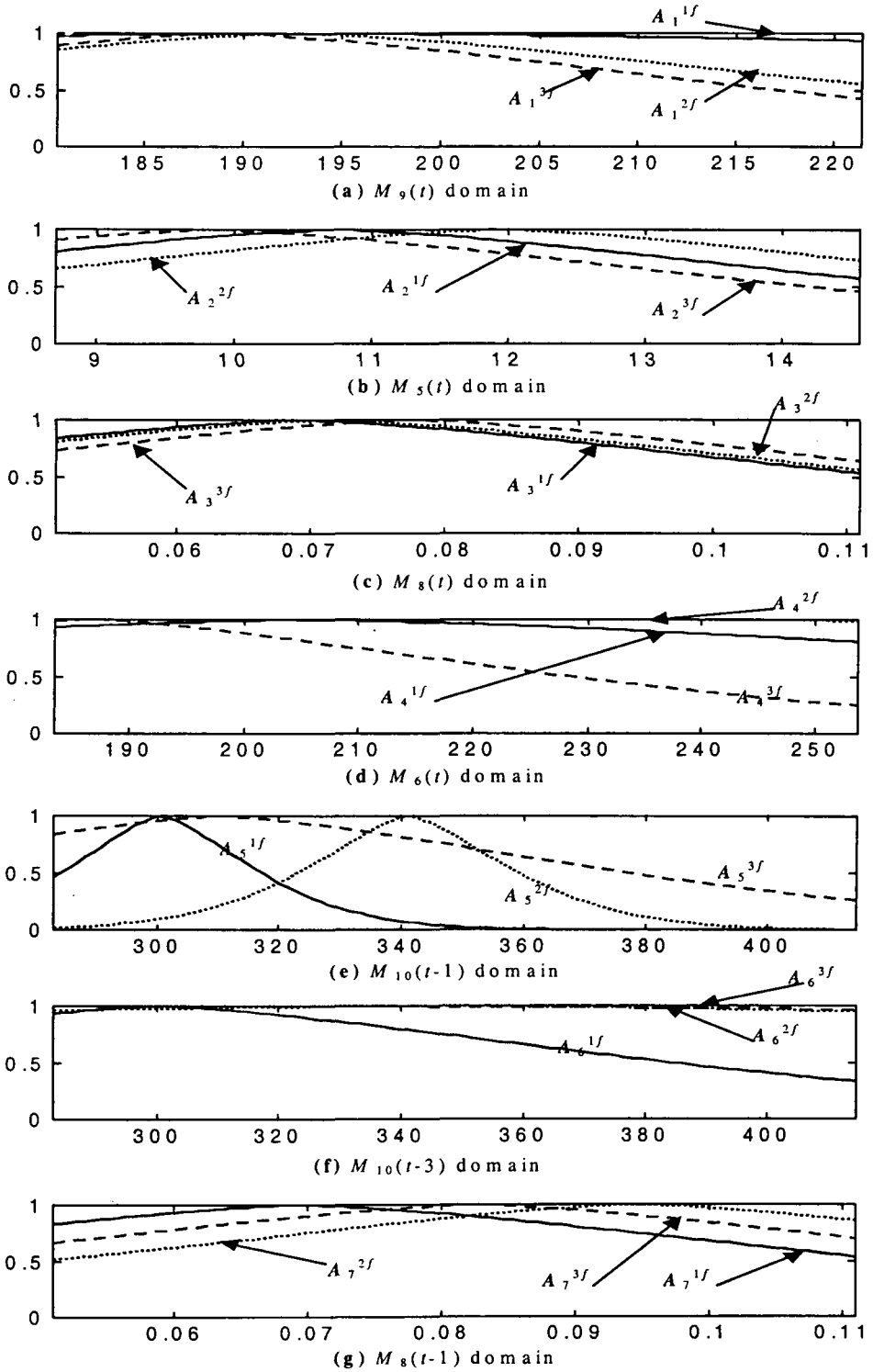


Fig. 6.45: Premise membership functions for class II GFM rules, of $M_{10}(t)$ variable

The output of the model and the actual values are plotted in Fig.6.46(a). Figure 6.46(b) shows the model error in per unit.

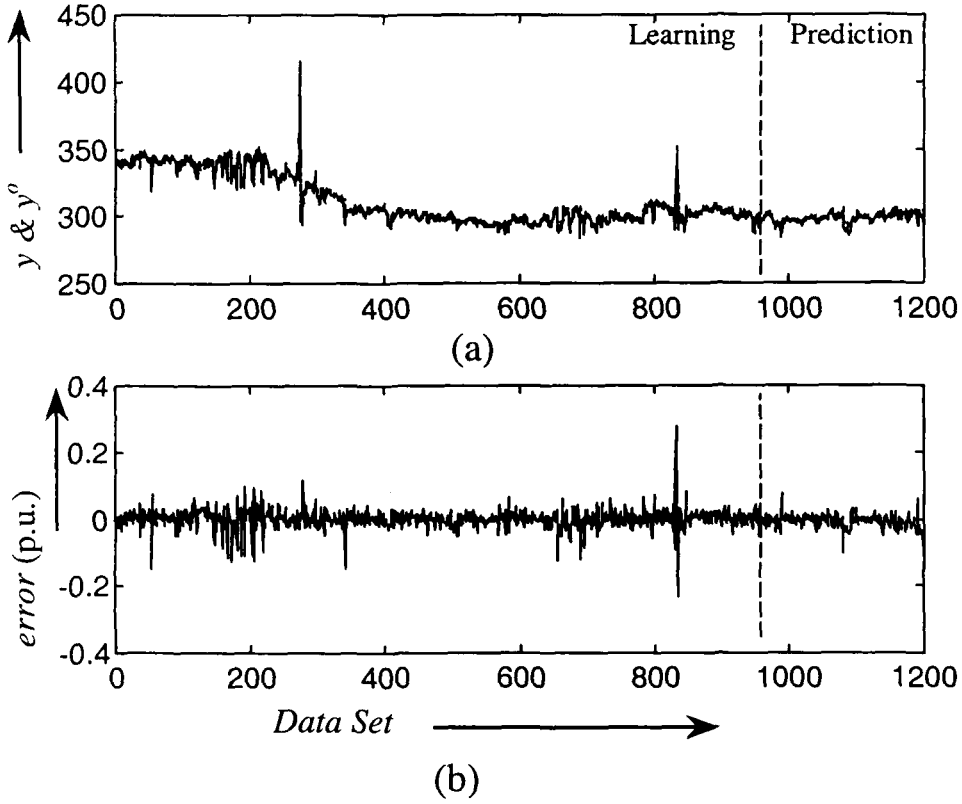


Fig.6.46: Plots of (a) y (————, solid line) and y^o (- - - - - - -,dashed line) (b) *model error* for class II GFM of $M_{10}(t)$.

It may be observed that in several figures (Figs. 6.15, 6.19,6.21, 6.25, 6.33, 6.35, and 6.45) the membership function of fuzzy variable are almost overlapping. This indicates that the corresponding fuzzy sets in the rules have the similar sense, i.e., the same fuzzy sets are appearing in different rules.

Among the controlled variables $C_1(t)$-$C_{10}(t)$, the variables $C_4(t)$, $C_5(t)$ and $C_9(t)$ have the arrangement for fine regulation besides the step change facility, which is provided for all the controlled variables. It can be seen from the error plot of any controlled variable that the large spikes occur around zero error at the instant of step change made in that particular controlled variable during the period of learning and prediction (See for example, Figs.6.9 and 6.15). Other than the instant of step change, the error comes to almost zero for the controlled variables not having regulation facility (See

for example, Fig.6.9). On the other hand, very small spikes appear around zero error for those controlled variables having regulation mechanism (See for example, Fig.6.15).

The plots of model error of the monitoring variables show small spikes around zero error during the periods of learning and prediction period, for all variables except for the variable Delta P across the Fractionator, i.e. M_8 (t). The model for $M_8(t)$ does not track the actual value during the prediction. The instantaneous effect of disturbance on variables $M_5(t)$, $M_8(t)$, $M_9(t)$, $M_{10}(t)$, caused by forcing the variable, Compressor Amperage, i.e. $M_6(t)$ to zero for a moment, can be noticed from the model outputs and plots of error for these variables(See the corresponding Figures). A very delayed effect can be seen on the variables $M_2(t)$, $M_3(t)$, and $M_7(t)$. The variables $M_1(t)$, and $M_4(t)$ bear no effect of disturbance caused by the variable, Compressor Amperage.

Discussion

When we consider the fractionator (without any safety measure) alone as a plant with vapor as input and its condensed part as the output, then it would behave like a linear plant under a certain range of operation, theoretically. But, FCCU consists of RRS and Fractionator. The variables associated with RRS and fractionator are interacting as shown in Table 6.3. This is because of high-pressure line connecting the RRS with the fractionator. If there is any disturbance in the fractionator, this will lead to a drastic change in pressure in the connecting line. While collecting the data of FCCU a disturbance is introduced by shutting the compressor on the fractionator side (See Fig. 6.38). The effect can be seen on the variables called Catalyst circulation rate of RRS (See Fig.6.36), Delta P across fractionator (See Fig.6.42), Side-Draw off temperature(See Fig.6.42), and Bottom vapor temperature (See Fig. 6.44).

6-5 Conclusions

This chapter deals with the modeling of FCCU to demonstrate the applicability of the proposed structure identification and learning approaches of previous chapters. First, a brief overview of FCCU consisting of RRS and Fractionator is presented followed by the demarcation of controlled and manipulated variables. Second, using fuzzy curve approach, the input selection as well as determination of class II GFM rules are carried out for all variables and the parameters of the models are then learned through LSE and GD learning. Since class II GFM gives the best performance for all the Examples of chapter 2, we have not considered the rest of the models. As in chapter 5, the rules are simplified by eliminating variables whose membership functions are nearly "1" over the range of operation. It is found that very few models require further learning by either GA hybrid or SA hybrid, as they do not achieve the targeted performance with the local learning itself. The results of this chapter culminate in the advance made in the modeling of a practical plant.

Chapter 7

CONCLUSIONS AND SUGGESTIONS FOR FURTHER WORK

7-1 Introduction

An important aspect of soft computing is that its constituents complement each other. Thus by adopting soft computing methodology, the deficiency in one technique can be complemented by another. For example, the model obtained after learning the RBF neural network, even though not interpretable, can be translated to human interpretable form by representing it in the form of fuzzy rules. The thesis looks at soft computing approaches for the modeling of complex dynamic systems as soft computing offers an olive branch to the evils of these systems. The application of soft computing in this thesis has resulted in advances in the components of modeling like input selection, structure determination and model learning. These advances appearing in different chapters are briefly described along with the conclusions drawn on them.

7-2 Conclusions

The input selection is the main concern of chapter 2. Takagi and Sugeno [**Takagi'85**] have proposed a method for input selection. For n input candidates and 2^p

maximum partitions, the maximum number of fuzzy systems, to be tested for input variable identification, by their method is of the order of $\mathcal{O}(n^2 p)$. Later on Sugeno-Yasukawa **[Sugeno'93]** have proposed another method by de-linking it from fuzzy partitioning. In this case the maximum number of fuzzy systems to be tested is $\mathcal{O}(n^2)$. However, for a system with a large number of input candidates, building such a large number of fuzzy test models is impractical. To solve this problem, the concept of Approximate Fuzzy Data Model (AFDM) is introduced and applied on dynamic systems. AFDM constitutes rules corresponding to each point (x_{ik} -y_k) in x_i -y plane (x_i is an input candidate). Aggregation of all these rules is the fuzzy curve $y^o(x_i)$. To evaluate the significance of input candidates, we have defined an index c_i and proposed new criteria for input selection. . For dynamic systems, we evaluate the significance of the past inputs, the past outputs and the current inputs with respect to current outputs. The computational complexity of this approach is linear with respect to the number of input candidates, i.e., $\mathcal{O}(n)$.

In chapter 3, a Generalized Fuzzy Model (GFM) is proposed. For this, conditions under which Takagi Sugeno (TS) Model and Compositional Rule of Inference (CRI) form of fuzzy model are equivalent have been evolved. Using the interpolation property on these models, the GFM is derived. To make GFM adaptive, a Generalized-Radial Basis Function (GRBF) network is proposed. It is the result of further generalization of the work of Jang **[Jang'93]** and Hunt, et al., **[Hunt'96]**. In this chapter, the issue of normalized network versus non-normalized network in the framework of GFM is tackled, effects of normalization of the Basis Functions in GFM are discussed and symmetric

property of GRBF network with regard to error surface is proved. The symmetry property has a great role to play in the choice of initialization of parameters and also the solution.

Next, the choice of shapes of the membership functions, the number of rules used and the inference mechanism on the performance evaluation of GFM are considered. It is noted that the effect of changing the shape of the membership functions and the number of rules is significant over that of the inference mechanism, but the size of rule base affects the computational efforts. For real-time applications, optimization of shapes of the membership functions and the number of rules is required for any fuzzy reasoning method. In this context, ANFIS of Jang **[Jang'93a]** using Gaussian membership function for premise variable poses a constraint on the FIS. However, most of the membership functions varying from triangular to trapezoidal are not in closed form suitable for differentiation, which is the prime requisite during the Gradient Descent (GD) learning. So a generalized Gaussian form of membership function, which caters to a wide range of membership functions, is proposed. Thus we have the Generalized Adaptive Neuro-Fuzzy Inference System (GANFIS), catering to the class I and the class II GFM, implemented through GRBFN.

Structure Identification of GFM is dealt in chapter 4. It consists of the selection of number of rule nodes in GRBF network and initialization of parameters of GFM. There is no generalized approach for the determination of an optimal rule set .in the literature. The optimum number of rules is arranged at appropriate positions in the fuzzy space of arbitrary shape, since rules do not cover the rectangular shapes in the input-output hyperspace $(x \times y)$. The minimum and maximum number of fuzzy rules is evaluated using a heuristic based on the idea that a fuzzy model will interpolate between the

maxima and minima of fuzzy curve. In order to obtain an optimal number of fuzzy rules, by making a trade off between complexity and performance, a Structure Identification Criterion (*SIC*) is defined. This optimality criterion trades off between the performance and the computational complexity of the model. A validity measure of clusters is applied for a specified number of clusters (rules). This measure is defined by the ratio of compactness to the separation of clusters. It is proposed that clustering should be carried out in the input-output hyperspace $(x \times y)$ **[Azeem'98]** instead of only in the input hyperspace x. Two types of clustering techniques, viz., Fuzzy C-means clustering and Modified Mountain clustering are used. In the Modified Mountain clustering it is presumed that every data point has a potential to become a cluster center instead of grid points formed by equidistant partitioning in each dimension of the $(x \times y)$ hyperspace **[Yager'94]**. A comparative performance of three different methodologies, viz., fuzzy-curve, Fuzzy C-means clustering and Modified Mountain clustering is shows that the initialization of GRBF network by Modified Mountain clustering yields model with the best performance with fewer rules. The fewer rules, which are decided by a heuristic based on fuzzy curve, avoid the over fitting of the data in the learning phase.

In chapter 5, Learning paradigm of GRBF is addressed and a solution is derived by the hybridization of the existing methodologies, namely Gradient Descent (GD), Genetic Algorithms (GA) along with Least Square Estimate (LSE). For adaptation of GFM via GRBF network, the estimation framework is formulated in the form of Least Square Estimate (LSE) technique. The LSE is implemented with stable-state Kalman filter for the evaluation of regression parameters in the output link, which represent the consequent part of GFM rule. GD learning is applied for the learning of basis function node

parameters and also for the learning of weights between the basis function layer and the normalizing layer of GRBF network. These weights represent index of fuzziness for consequent regression in GFM rule. The hybrid of LSE and GD termed local learning has been explored in this chapter. During the process of adaptation, GRBF network may end up in local minima, which is a very peculiar property of GD technique. To overcome the problem of local minima, use of global learning techniques, like Genetic Algorithms (GA) and Simulated Annealing (SA), is made in combination with local learning algorithms yielding GA and SA hybrids respectively. These hybrid techniques are applied only on the Stock Market data **[Sugeno'93]** to achieve the targeted performance, which could not be achieved through local learning.

In chapter 6, a case study of Multi-Input and Multi-Output (MIMO) system is carried out. This case study is for the Fluidized Catalytic Cracking Unit (FCCU) of a petroleum refinery in INDIA. The variables involved in FCCU are categorized as controlled and manipulated and are fitted with class II GFMs which give the best performance among all the models. The rules of GFM are decided by a heuristic based on the fuzzy curve approach and the initial parameters of class II GFM rules are obtained from the. Modified Mountain clustering. Then local learning is applied for the fine tuning the parameters. Most of the fine tuned models are found to achieve the targeted performance.

7-3 Contributions of the thesis

The important contributions made in the thesis are listed as under:

1. Interpretation of the Fuzzy Curve as Unconditional Mean.
2. Criteria for input selection

3. Proposition of generalized membership function for premise variable.
4. Formulation of GFM
5. Proposition of Generalized RBF network, functionally equivalent to GFM.
6. Specification of an optimum rule set applied for fuzzy clustering of input-output hyperspace.
7. Reformulation of GFM in LSE framework for the subsequent use of Kalman filter.
8. Experimentation of hybrid learning techniques to avoid the local minima.
9. Demonstration of various theoretical results of the thesis on a real life process.

7-4 Limitations of the proposed approaches

Barring the theoretical developments, some loopholes develop in the implementations due to inadequacies of the underlying theory. While using the criterion 1 need arises for a threshold which has an effect on the choice of inputs. Determination of optimum number of clusters is dependent on the choice of a few parameters. Any improper selection would have a deteriorating effect on the performance of the model chosen. Likewise, initial values of the parameters will have a role in the aftereffects that accrue out of injudicious selection. The learning rates affect drastically the speed of convergence of learning algorithms. There is no cut and paste solution to these limitations, which finally result in poor performance of models, though certain guidelines have been provided from the experience on the algorithms in chapter 5, they may not always work out on different problems.

7-5 Suggestions for further work

We have categorized all the models into four classes depending on the type of operators employed on the membership functions of rules while computing the output. In this thesis, we have used class I and class II models of CRI and GFM forms only along with TS model which is not amenable to any class. The other two classes of models need be explored as a further study.

Recently, some new results have appeared in the literature on robust clustering for an unknown number of clusters from noisy data **[Frigui'96]** and clustering by competitive Agglomeration (CA) **[Frigui'97].** These are found to be quite impractical but one may have a second look for their desirability for suitable number of cluster formation.

On genetic algorithmic front, several new results have to come light. These results aim at time efficiency by using probabilistic crowding with probabilistic replacement of populations. These may be incorporated into our GA hybrid to derive extra mileage.

The thesis has dealt with mainly the modeling part, control part could not be tackled for want of time. But the proposed approaches are still useful for the design of controller by inverse modeling of systems.

REFERENCES

[Akaike'74] H. Akaike, "A New Look at the Statistical Model Identification", *IEEE Trans. on Automat. Control*, Vol. 19, pp. 716-723. 1974.

[Azeem'98a] M. F. Azeem, M. Hanmandlu, N. Ahmad, "An Application of Fuzzy Clustering in Dynamic Fuzzy Modeling", *Proc. 3rd Intl. Conf. on Appl. of Fuzzy systems and Soft Computing ICAFS-98,* Siegen, Germany, October 5-7, 1998, pp. 143-155.

[Azeem'98b] M. F. Azeem, M. Hanmandlu, N. Ahmad, "A New Criteria for Input Variable Identification of Dynamical Systems", *Proc. IEEE Region 10 Intl. Conf., TENCON'98,* New Delhi, India, December 17-19, 1998, pp. 230-233..

[Azeem'99] M. F. Azeem, M. Hanmandlu, N. Ahmad, "Modified Mountain Clustering in Dynamic Fuzzy Modeling", *2nd Intl. Conf. on Infor. Tech.,* Bhubaneswar, India, December *20-23*, 1999.

[Back'95] T. Back, and H.P. Schwefel,. "Evolution Strategies I: Variants and their computational implementation", In G. Winter, J. P'eriaux, M. Gal'an, and P. Cuesta, **Genetic algorithms in engineering and computer science**, Chichester: John Wiley, pp. 111-126, 1995.

[Beniam'94] M. Beniam, "On Functional Approximation with Normalized Gaussian Units", *Neural Computation*, Vol. 1, pp. 319-333, 1994.

[Billing'95] S.A. Billing and G.L. Zheng, "Radial Basis Function Network Configuration Using Genetic Algorithms", *Neural Networks*, Vol. 8, No. 6, pp. 877-890. 1995.

[Box'70] E. P. Box and G. M. Jenkins, **Time Series Analysis, Forecasting and Control,** San Francisco: Holden Day, 1970.

[Broomhead'88] D.S. Broomhead and D. Lowe, "Multivariable Functional Interpolation and Adaptive Networks", *Complex Systems*, vol. 2, pp. 321-355, 1988.

[Chao'96] C.T. Chao, Y.J. Chen, C.C. Teng, "Simplification of Fuzzy-Neural Systems Using Similarity Analysis", *IEEE Trans. on Systems, Man and Cybernetics-Part B: Cybernetics*, Vol. 26, No. 2, pp.344-353, April 1996.

[Chen'91] S. Chen, C.F.N. Coan and P.M. Grant, 'Orthogonal Least square learning Algorithm for Radial Basis Function Networks", *IEEE Trans. on Neural Networks*, Vol. 2, No. 2, ,pp. 302-309, 1991.

[Chin'98] L. Chin, D.P. Mittal, B. Low, N. Lim, "Neuro-Fuzzy techniques for Airborne Multi-Target Tracking", *Proc. IEEE Region 10 Intl. Conf., TENCON'98,* New Delhi, India, December 17-19, 1998, pp. 195-199.

[Davis'89] L. Davis, "Adapting operator probabilities in genetic algorithms", *Proc. of the 3rd International Conference on Genetic Algorithms*, pp. 60-69, 1989.

[Davis'91] L. Davis, "**The handbook of genetic algorithms**", New York. Van Nostrand Reinhold, 1991.

[Deshpande'85] P. B. Deshpande, "Distillation dynamics and control", *Instrument Society of America*, pp281, 1985.

[Donkelaar'98] V. Donkelaar, T. Edwin, S.C.P. Heuberger, V.D. Hof and M. J. Paul " Identification of a fluidized catalytic cracking unit an orthonormal basis function approach" *Proc. of American Control Conf.*, Vol.3, pp 1914 -1917, June 1998.

[Frigui'96] H. Frigui and R. Krishnapuram, "A robust Algorithm for Automatic Extraction of an unknown number of Cluster from Noisy Data", *Recognition Letters,* Vol.17, No. 12, pp.1223-1232, October 1996

[Frigui'97] H. Frigui and R. Krishnapuram, "Clustering by Competitive Agglomeration", *Pattern Recognition*, Vol.30, No. 7, pp.1109-1119, July 1997.

[Fukuda'92] T. Fukuda Y. Sunahara, "Identification of Vaguely Dependent Parameters for A Class Of Fuzzy Stochastic Systems", *Proc. on Fuzzy Systems Intl Conf.* 1992, pp. 1419-1426.

[Goldberg'89] D. E. Goldberg, "**Genetic algorithms in search, optimization, and machine learning**", Reading, MA: Addison-Wesley, 1989.

[Goldberg'91] D. E. Goldberg, "Real-coded genetic algorithms, virtual alphabets, and blocking", *Complex Systems*, Vol. 5, pp139-167, 1991.

[Grefenstette'85] J. Grefenstette, R. Gopal, B. Rosmaita, and D. Van Gucht, "Genetic algorithms for the traveling salesman problem", *Proc. of an Intl. Conf. on Genetic Algorithms and Their Application,* 1985, pp. 160-165,.

[Gupta'98] M. M. Gupta, "Fuzzy-Neural approach in the Development of Cognitive Robotic Systems", *Proc. IEEE Region 10 Intl. Conf., TENCON'98,* New Delhi, India, December 17-19, 1998, pp. 189-194.

[Hinton'87] G. E. Hinton, and S. J. Nowlan, "How learning can guide evolution", *Complex Systems*, Vol. 1, pp. 495-502, 1987.

[Hong'96] T.P. Hong, and C.Y. Lee, "Induction of fuzzy rules and membership functions from training examples", *Fuzzy Sets and Systems*, Vol. 84, No. 1, pp. 33-47, 1996.

[Hunt'96] Kenneth J. Hunt, Ronald Haas and Roderick Murray-Smith, "Extending Functional Equivalence of Radial Basis Function Networks and Fuzzy Inference Systems", *IEEE Trans. on Neural Networks*, Vol. 7, No. 3, pp.776-781, May. 1996.

[Ibaraki'97] T. Ibaraki, "Combinations with other optimization methods", In T. Back, D. B. Fogel, and Z. Michalewicz (Eds.), **Handbook of Evolutionary Computation (Section D3),** New York: Oxford University Press, 1997.

[Ishibuchi'95] H. Ishibuchi, K. Nozaki, N. Yamamoto and H. Tanaka, "Selecting Fuzzy If-Then Rules for Classification Problems Using Genetic Algorithms", *IEEE Trans. on Fuzzy Systems*, Vol.3, No. 3, pp. 260-270, Aug. 1995.

[Jang'93a] J.-S. R. Jang and C.-T. Sun, "Functional Equivalence Between Radial Basis Function Networks and Fuzzy Inference Systems", *IEEE Trans. on Neural Networks*, Vol.4, No. 1, pp.1139-1150, Jan. 1993.

[Jang'93b] J.-S. R. Jang, "ANFIS: Adaptive Network Based Fuzzy Inference Systems", *IEEE Trans. on Systems, Man and Cybernetics*, Vol.23, No. 3, pp. 650-684, May/June. 1993.

[Jones'89] R.D. Jones, Y.C. Lee, C.W. Barnes, G.W. Flake, K. Lee, S.P. Lewis, and S. Qian, " Function Approximation and Time-Series Prediction with Neural Networks", *Proc. IEEE Conf. on Neural Networks*, 1989.

[Jones'92] T.A. Jones, B.A. Foss, " Representing and learning unmodeled dynamics with neural network memories", *Proc. Amer. Contr. Conf.*, Chicago, IL., June 1992, pp. 3037-3043.

[Karr'91a] Chuck Karr, "Genetic Algorithms for Fuzzy Controller", *AI Expert*, , pp. 26-33, Feb. 1991.

[Karr'91b] Charles L. Karr, "Design of an Adaptive Fuzzy Logic Controller Using a Genetic Algorithm", *Proc. of 4th Genetic Algorithms Conf.*, 1991, pp. 450-457.

[Karr'93] C.L. Karr, E.J. Gentry, "Fuzzy control of PH using Genetic algorithms.", *IEEE Trans. on Fuzzy Systems*, Vol. 1, No.1, pp. 46-53, 1993

[Kirkpatrick'83] S. Kirkpatrick, CD Gelatt and MP Veccchi, "Optimization by Simulated Annealing", *Science,* Vol. 220, No. 4598, pp. 671-680, May 1983.

[Kosko'97] B. Kosko, "**Fuzzy Engineering**", Prentice-Hall Inc., New Jersey, pp. 80-89, 1197.

[Lengari'97] R. Lengari, L. Wang and J. Yen, "Radial Basis Function Networks. Regression Weights, and Expectation-Maximization Algorithm", *IEEE Trans. on Systems., Man, Cybernetics-Part A: Systems Humans*, Vol.27, No. 5, pp.613-623, 1997.

[Leontaritis'87] I.J. Leontaritis and S.A. Billings, "Model Selection and Validation Methods for nonlinear Systems", *Intl. Jour. Contr.*, Vol. 45, No. 1, pp 311-341,1987.

[Light'92] W.A. Light, "Some Aspects of Radial Basis Function Approximation", *Approximation Theory, Spline Function Application*, Vol. 356, pp.163-190, 1992.

[Lin'95] Y. Lin and G.A. Cunningham III "A New Approach to Fuzzy-Neural System Modeling" *IEEE Trans. on Fuzzy Systems*, Vol. 3, No.2, pp.190-198, 1995.

[Liska'94] J. Liska and S. S. Melsheimer, "Complete Design of Fuzzy Logic Systems Using Genetic Algorithms", *Proc. of Intl Conf. on Fuzzy Systems*, 1994, pp. 1377-1382.

[Lobo'97] F. Lobo, and D. E. Goldberg, "Decision making in a hybrid genetic algorithm", *Proc. of IEEE Intl Conf. on Evolutionary Computation*, 1997, pp. 121-125.

[Lotfi'96] A. Lotfi and A.C. Tsoi, "Learning Fuzzy Inference Systems Using an Adaptive Membership Function Scheme", *IEEE Trans. Systems Man And Cybernetics-Part B: Cybernetics*, Vol. 26, No. 2, pp. 326-331, April 1996.

[Lu'91] Y. Z. Lu, and M. He, "An expert control system for the fluidized catalytic cracking unit in refinery" *Expert System Application in Advanced Control* , Printout from UOP, 1991.

[Machado'92] R. J. Machado and A. F. da Rocha, "Evolutive Fuzzy Neural Networks", *Proc. of Intl. Conf. on Fuzzy Systems*, 1992, pp. 493-500

[Mamdani'74] E.H. Mamdani, "Application of fuzzy algorithms for control of simple dynamic plant", *Proc. IEE*, Vol. 121, No. 12, December, pp. 1585-1588, 1974.

[Mamdani'77] E.H. Mamdani, "Application of Fuzzy Logic To Approximate Reasoning Using Linguistic Synthesis", *IEEE Trans. On Computers* , Vol. C-26, No. 12, pp. 1182-1191, December 1977.

[Mizumoto'88] M. Mizumoto, "Fuzzy Controls Under Various Fuzzy Reasoning Methods", *Information Science*, Vol. 45, pp. 129-151, 1988.

[Moody'88] J. Moody and C. Darken, "Learning with Localized Receptive Fields", *Proc. 1988 connectionist models summer school*, D. Touretzky, G. Hinton, and T. Sejnoeski, (Eds.) Carnegie Mellon University, Morgan Kaufmann Publishers, 1988.

[Moody'89] J. Moody and C. Darken, " Fast Learning in Neural Networks of Locally-Tuned Processing Units", *Neural Computation*, Vol. 1, pp. 281-294, 1989.

[Nauck'92] D. Nauck, F. Klawonn, R. Kruse, "Fuzzy sets, Fuzzy Controllers and Neural Networks", *Scientific journal of Humbolt-University of Berlin*, Series Medicine 41, No. 4, pp. 99-120, 1992.

[Nauck'94] D. Nauck, R. Kruse, "Choosing appropriate Neuro Fuzzy Models", *Proc. of Second European Congress of Intelligent Techniques and Soft Computing* (EUFIT'94), Aachen, Germany, 1994, pp. 552-557.

[Nauck'95] D. Nauck "Beyond Neuro-Fuzzy: Perspective and Directions", *Proc. of Third European Congress of Intelligent Techniques and Soft Computing* (EUFIT'95), Aachen, Germany, August 28-31'1995, pp. 1159-1164.

[Nauck'97] D. Nauck, R. Kruse, "Neuro-Fuzzy Systems for function Approximation", *Presented in 4th International Workshop on Fuzzy-Neuro Systems*, Soest, 1997.

[Nauck'98] D. Nauck, R. Kruse, " A Neuro-Fuzzy Approach to Obtain Interpretable Fuzzy Systems for Function Approximation ", *Proc. of IEEE Intl Conf. on Fuzzy Systems,* 1998 (FUZZ-IEEE'98), Anchorage, AK, May 4-9, 1998, pp. 1106-1111.

[Nozaki'97] K. Nozaki, H. Ishibuchi, and H. Tanaka, "A simple but powerful heuristic method for generating fuzzy rules from numerical data", *Fuzzy Sets and Systems*, Vol. 86, No. 3, pp. 251-270, 1997.

[Park'95] Y.-M. Park, et al., "A Self Organizing Fuzzy Logic Controller for Dynamic Systems Using a Fuzzy Auto-Regressive Moving Average (FARMA) Model", *IEEE Trans. on Fuzzy System,* Vol.-3, No.1, pp.75-82, Feb. 1995.

[Pedrycz'96] Witold Pedrycz, "Fuzzy Multimodels", *IEEE Trans. on Fuzzy Systems*, Vol. 4, No. 2, pp. 139-148, May 1996.

[Perneel'95] C. Perneel, J.-M. Themlin, J.-M. Renders, and M. Acheroy, "Optimization of Fuzzy Expert Systems Using Genetic Algorithms and Neural Networks". *IEEE Trans. on Fuzzy Systems*, Vol.3, No.3, Aug. 1995.

[Powell'89] D. Powell, S. S. Tong, and M. M. Skolnick, "EnGENEous domain independent, machine learning for design optimization", *Proc. of the Third Intl Conf.on Genetic Algorithms*, 1989, pp.151-159.

[Powell'90] M.J.D. Powell "Radial Basis Functions in 1990", *Adv. Numer. Anal.*, Vol. 2, pp. 105-210, 1990.

[Randelman'86] R.E. Randelman, G.S. Grest, "N-city travelling salesman problem: Optimization by simulated annealing", *Journal of Statistical Physics*, Vol. 45, pp. 885-890, 1986

[Shaefer'87] C. G. Shaefer, "The ARGOT strategy: Adaptive representation genetic optimizer technique", Genetic Algorithms and Their Applications: *Proc. of the Second Intl Conf. on Genetic Algorithms*, 1987, pp. 50-58.

[Shorten'94] R. Shorten and R. Murray-Smith, "On Normalizing Basis Function Networks", *Proc. of 4th Irish Neural Network Conf.,* Univ. College Dublin, September 1994

[Smith'85] D. Smith, "Bin packing with adaptive search", *Proc. of an Intl Conf. on Genetic Algorithms and Their Application*, 1985, pp. 202-206,.

[Sugeno'88] M. Sugeno and G.T. Kang "Structure Identification of Fuzzy Model" *Fuzzy Sets and Systems,* Vol. 28, pp.15-33, 1988.

[Sugeno'93] M. Sugeno, T. Yasukawa, "A Fuzzy logic based approach to qualitative modeling.", *IEEE Trans. on Fuzzy Systems*, Vol. 1, No.1, pp. 7-31, 1993.

[Sugiyama'88] Kenji Sugiyama, "Rule-Based Self Organizing Controller", **Fuzzy computing**, M.M. Gupta and T. Yamakawa(Eds.), Elsevier Science Publisher B.V.(North Holland), pp.341-353,1988.

[Takagi'85] T. Takagi and M. Sugeno "Fuzzy Identification of Systems and Its Application to Modeling and Control" *IEEE Trans on Systems, Man and Cybernetics*, Vol. 15 No.1, pp.116-132, Jan/Feb'1985.

[Tan'94] S. Tan, J. Hao and J. Wandewalle, "Identification of Nonlinear Discrete- Time Multivariable Dynamic Systems by RBF Neural Networks", *Proc. of IEEE Intl. Conf. on Neural Networks*, Vol. 5, 1994, pp. 3250-3255

[Venuto'79] P. B. Venuto, and E. T Habib, **Fluid catalytic cracking with zeolite catalysts**, Marcel Dekker Inc., 1979.

[Wang'92a] L.X. Wang and Jerry M. Mendel, "Generating Fuzzy rules by learning from examples", *IEEE Trans. on Systems Man and Cybernetics*, Vol. SMC-22, pp. 1414-1427, 1992.

[Wang'92b] L. X. Wang and J. M. Mendel, 'Fuzzy Basis Function, Universal Approximation, and Orthogonal Least square learning", *IEEE Trans. on Neural Networks*, Vol. 3, No. 5, ,pp. 807-814, 1992.

[Wang'95] Li-Xi Wang "Design and Analysis of Fuzzy Identifiers of Nonlinear Dynamic Systems", *IEEE Trans. on Automatic Control*, Vol-40, No.1, pp. 11-23, Jan'1995.

[Whitley'95] D. Whitley, "Modeling hybrid genetic algorithms", In G. Winter, J. P'eriaux, M. Gal'an, and P. Cuesta (Eds.), **Genetic algorithms in engineering and computer science,** pp. 191-201, Chichester John Wiley, 1995.

[Whittington'72] Whittington, E. L., Murphy, J. R. and Lutz, I. H. "Gasoline from FCCU" *Journal of American Chemical Society*, Vol.17, No.3, sec. B66 , 1972.

[Xie'91] X.L. Xie and G. Beni, "A Validity Measure for Fuzzy Clustering", *IEEE Trans. on Pattern Anal. Machine Intell.*, Vol. 13, No. 8, pp. 841-847, 1991.

[Xu'87] C.W. Xu and Y.Z. Lu, " Fuzzy Model Identification and Self-Learning for Dynamic System" *IEEE Trans. on Systems Man and Cybernetics*, Vol.17, No. 4, pp. 683-689, July/Aug. 1987.

[Yager'94] R. R. Yager and D. P. Filev, **Essential of Fuzzy Modeling and Control**, John Wiley & Sons, 1994

[Yager'92] R. R. Yager and D. P. Fliev, "Identification of Nonlinear Systems by Fuzzy Models", *Proc. on Fuzzy Systems Intl. Conf.*, 1992, pp. 1401-1408.

[Yingwei'96] L. Yingwei, N. Sundararajan and P. Saratchandran, "Adaptive Nonlinear System Identification Using Minimal Radial Basis Function Neural Networks", *Proc. of IEEE Intl. Conf. On Acoustic, Speech and Signal processing*, Vol. 6, pp. 3521-3524, 1996.

[Yoshikawa'94] T. Yoshikawa, T. Furuhashi and Y. Uchikawa, "A Fuzzy Modeling of very Large Scale System Using Genetic Algorithm and a Multiple -Representing Method", *Proc. of Fuzzy Systems Intl Conf.*, 1994, pp. 1895-1898.

[Zadeh '73] Lotfi A. Zadeh, " Outline of a new Approach to the Analysis of Complex Systems And Decision Processes", *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 3, No. 1, pp.28-44, 1973.

[Zeng'94] X. J. Zeng, and M.G. Singh, "Approximate theory of Fuzzy systems-SISO.", *IEEE Trans. on Fuzzy Systems*, Vol. 2, No.2, pp. 162-176, 1994.

[Zeng'95a] X.-J. Zeng and M. G. Singh, 'Approximation Accuracy Analysis of Fuzzy Systems as Function Approximators", *IEEE Trans. on Fuzzy Systems* Vol.-4,No.1, Feb. 1996, pp. 44-63

[Zeng'95b] X.-J. Zeng and M. G. Singh, 'Approximation Theory of Fuzzy Systems-MIMO Case" *IEEE Trans. on Fuzzy Systems,* Vol.-3,No.2 pp.219-235, May'1995.

[Zeng'96a] X.-J. Zeng and M. G. Singh, "A Relationship Between Membership Functions And Approximation Accuracy in Fuzzy Systems", *IEEE Trans. on Systems Man And Cybernetics-Part B: Cybernetics*, Vol. 26, No.1, pp. 176-180, February 1996

[Zeng'96b] X.-J. Zeng and M. G. Singh, " Approximation Properties of Fuzzy Systems Generated By the Min Inference", *IEEE Trans. on Systems Man And Cybernetics-Part B: Cybernetics*, Vol. 26, No. 1, pp.187-193, February. 1996

[Zeng'96c] X.-J. Zeng and M. G. Singh, Decomposition Property of Fuzzy Systems and Its Applications", *IEEE Trans. on Fuzzy Systems*, Vol. 4, No.2, pp. 149-165, May 1996.

Appendix A

To define the general form of basis function $\varphi^k(\boldsymbol{x})$ we need a few matrix operations defined in the following.

Definition1: For matrix $\boldsymbol{A} = \{a_{ij} : a_{ij} \in R \ , \ i = 1,\ldots,n \ ; j = 1,\ldots,m\}$ the operation $\|\boldsymbol{A}\|$ results in a matrix $\boldsymbol{B} = \{b_{ij} : b_{ij} \in R^+ \ , \ i = 1,\ldots,n \ ; j = 1,\ldots,m\}$ where

$$b_{ij} = a_{ij} \quad \text{if } a_{ij} \geq 0$$
$$b_{ij} = -a_{ij} \quad \text{if } a_{ij} < 0$$

Definition2: For matrices $\boldsymbol{A} = \{a_{ij} : a_{ij} \in R^+ \ , \ i = 1,\ldots,n \ ; j = 1,\ldots,m\}$ and $\boldsymbol{B} = \{b_{ij} : b_{ij} \in R^+ \ , \ i = 1,\ldots,n \ ; j = 1,\ldots,m\}$ the operation $\boldsymbol{A}.\hat{}\,\boldsymbol{B}$ results in a matrix $\boldsymbol{C} = \{c_{ij} : c_{ij} \in R^+ \ , \ i = 1,\ldots,n \ ; j = 1,\ldots,m\}$ where $c_{ij} = a_{ij}^{\,b_{ij}}$. If $\boldsymbol{B} = \{b : b \in R^+ \}$ is scalar then $c_{ij} = a_{ij}^{\,b}$

Definition3: For vector $\boldsymbol{a} = \{a_i : a_i \in R \ , \ i = 1,\ldots,n \}$ the operation $\mathbf{DM}(\boldsymbol{a})$ results in a matrix $\boldsymbol{B} = \{b_{ij} : b_{ij} \in R \ \ , \ i = 1,\ldots,n \ ; j = 1,\ldots,n\}$ where

$$b_{ij} = a_i \quad \text{if } i = j$$
$$b_{ij} = 0 \quad \text{if } i \neq j$$

B-1 Proof for C-means Clustering

The problem of minimization of objective function J_{cy} {give eqn.(4.10)}subjected to the constraint specified by eqn.(4.8), is solved by minimiz constraint free objective function J_c defined as:

$$J_c = J_{cy} + \sum_{j=1}^{M} \lambda_j \left(\sum_{k=1}^{m} \mu_{jk} - 1 \right) \quad \ldots$$

where, λ_j ,(j= 1, 2, . . . , M) are Langrangian multipliers

By taking the partial derivatives of J_c with respect to $\bar{c}_{ky}$, λ_j and μ_{jk}, yiel solution for the problem:

B-1.1 Partial Derivative of J_c with respect to $\bar{c}_{ky}$

The partial derivative of J_c with respect to $\bar{c}_{ky}$ is:

$$\frac{\partial J_c}{\partial \bar{c}_{ky}} = -2 \sum_{j=1}^{M} \mu_{jk}^{q} . \left(\bar{x}_{jy} - \bar{c}_{ky} \right) \quad \ldots(B\text{-}2)$$

Equating eqn.(B-2) to zero leads to,

$$\frac{\partial J_c}{\partial \bar{c}_{ky}} = 0$$

$$\Rightarrow \quad -2 \sum_{j=1}^{M} \mu_{jk}^{q} . \left(\bar{x}_{jy} - \bar{c}_{ky} \right) = 0.$$

$$\Rightarrow \quad -\sum_{j=1}^{M} \mu_{jk}^{q} . \bar{x}_{jy} + \sum_{j=1}^{M} \mu_{jk}^{q} \bar{c}_{ky} = 0.$$

$$\bar{c}_{ky} = \frac{\sum_{j=1}^{M} \mu_{jk}^{q} . \bar{x}_{jy}}{\sum_{j=1}^{M} \mu_{jk}^{q}} \qquad \text{...(B-3)}$$

B-1.2 Partial Derivative of J_c with respect to λ_j

The partial derivative of J_c with respect to λ_j is:

$$\frac{\partial J_c}{\partial \lambda_j} = \left(\sum_{k=1}^{m} \mu_{jk} \right) - 1 \qquad \text{...(B-4)}$$

Equating eqn.(B-4) to zero leads to the following:,

$$\frac{\partial J_c}{\partial \lambda_j} = 0$$

$$\Rightarrow \quad \sum_{k=1}^{m} \mu_{jk} = 1 \qquad \text{...(B-5)}$$

Equation (B-5) is the given constraint to J_{cy}

B-1.3 Partial Derivative of J_c with respect to μ_{jk}

The partial derivative of J_c with respect to μ_{jk} is:

$$\frac{\partial J_c}{\partial \mu_{jk}} = \frac{\partial J_{cy}}{\partial \mu_{jk}} + \frac{\partial}{\partial \mu_{jk}} \left\{ \sum_{j=1}^{M} \lambda_j \left[\left(\sum_{k=1}^{m} \mu_{jk} \right) - 1 \right] \right\} \qquad \text{...(B-6)}$$

$$= q.\mu_{jk}^{(q-1)} d_{jk}^2 \left(\bar{x}_{jy}, \bar{c}_{ky} \right) + \lambda_j$$

equating eqn.(B-6) to zero leads to the following:

$$\frac{\partial J_c}{\partial \lambda_{jk}} = 0$$

$$\Rightarrow \quad q.\mu_{jk}^{(q-1)} d_{jk}^2 \left(\bar{x}_{jy}, \bar{c}_{ky} \right) + \lambda_j = 0$$

$$\Rightarrow \quad \mu_{jk} = -\left(\frac{\lambda_j}{q} \right)^{\frac{1}{(q-1)}} \cdot \left(\frac{1}{d_{jk}^2 \left(\bar{x}_{jy}, \bar{c}_{ky} \right)} \right)^{\frac{1}{(q-1)}} \qquad \text{...(B-7)}$$

To fulfill the constraint.(B-5),

$$\Rightarrow \quad \mu_{jk} = \frac{\mu_{jk}}{\sum_{k=1}^{m} \mu_{jk}} \qquad \text{...(B-8)}$$

In view of eqn.(B-8), eqn.(B-7) can be written as

$$\Rightarrow \quad \mu_{jk} = \frac{\left(\frac{1}{d_{jk}^2 \left(\bar{x}_{jy}, \bar{c}_{ky} \right)} \right)^{\frac{1}{(q-1)}}}{\sum_{k=1}^{m} \left(\frac{1}{d_{jk}^2 \left(\bar{x}_{jy}, \bar{c}_{ky} \right)} \right)^{\frac{1}{(q-1)}}} \qquad \text{...(B-9)}$$

B-2 Cluster Validity Measure, S

Cluster validity measure S is defined as the ratio of compactness to separation. The terms compactness and separation are expressed in the following:

B-2.1 Compactness of clusters

Consider a fuzzy C-means clusters of the normalized data set $\bar{x}_{jy} = \left(\left\{\bar{x}_j, \bar{y}_j\right\} \forall \quad j = 1, \ldots, M\right)$, obtained from the given data set $\mathcal{D}_M = \left\{x_j, y_j\right\}_{j=1}^{M}$, with the centroid $\bar{c}_{ky}\ (k = 1, \ldots, m)$ of each cluster and $\mu_{jk}\ (j = 1, \ldots, M;\ k = 1, \ldots, m)$ as the fuzzy membership of data point j belonging to cluster k. The *fuzzy deviation* of data $\bar{x}_{jy}$ from the cluster $\bar{c}_{ky}$ is defined as

$$\mathcal{d}_{jk} = \mu_{jk} d_{jk}\left(\bar{x}_{jy}, \bar{c}_{ky}\right) \qquad \text{...(B-10)}$$

where, $d_{jk}\left(\bar{x}_{jy}, \bar{c}_{ky}\right) = \sqrt{d_{jk}^2\left(\bar{x}_{jy}, \bar{c}_{ky}\right)} = \left\|\left(\bar{x}_{jy} - \bar{c}_{ky}\right)\right\|$

The *fuzzy cordiality* of cluster $\bar{c}_{ky}$ is defined as

$$M_k = \sum_{j=1}^{M} \mu_{jk} \qquad \text{...(B-11)}$$

Note that $\sum_{k=1}^{m} M_k = M$ is the total number of data

For each cluster, the summation of squares of the fuzzy deviations of each data from the centroid, denoted by ρ_k, is called the *variation* of cluster 'k'

$$\rho_k = \sum_{j=1}^{M} \mathcal{d}_{jk}^2 = \sum_{j=1}^{M} \mu_{jk}^2 d_{jk}^2\left(\bar{x}_{jy}, \bar{c}_{ky}\right) \qquad \text{...(B-12)}$$

The summation of variation of all clusters, denoted by ρ, is called the *total variation* of all data set $\bar{x}_{jy}$ with respect to fuzzy C-means clusters, i.e.,

$$\rho = \sum_{k=1}^{m} \rho_k = \sum_{k=1}^{m}\sum_{j=1}^{M} \mathscr{A}_{jk}^2 = \sum_{k=1}^{m}\sum_{j=1}^{M} \mu_{jk}^2 d_{jk}^2 \left(\bar{x}_{jy}, \bar{c}_{ky}\right) \qquad \text{...(B-13)}$$

The ratio of the total variation to the size of data set is called the *compactness*, denoted by Π

$$\Pi = \frac{\rho}{M} = \frac{\sum_{k=1}^{m}\sum_{j=1}^{M} \mu_{jk}^2 d_{jk}^2 \left(\bar{x}_{jy}, \bar{c}_{ky}\right)}{M} \qquad \text{...(B-14)}$$

B-2.2 Separation of clusters

The square of the minimum distance out of all possible distances among all the cluster centroids is called the *separation*, denoted by ϖ, of the fuzzy C-means clusters i.e.,

$$\varpi = \min_{\substack{j,k \\ j \neq k}}^{m} \left\{ d_{jk}^2 \left(\bar{c}_{jy}, \bar{c}_{ky}\right) \right\} = \min_{\substack{j,k \\ j \neq k}}^{m} \left\| \bar{c}_{jy} - \bar{c}_{ky} \right\|^2 \qquad \text{...(B-15)}$$

The *validity function S* is therefore evaluated from

$$S = \frac{\Pi}{\varpi} = \frac{\sum_{k=1}^{m}\sum_{j=1}^{M} \mu_{jk}^2 d_{jk}^2 \left(\bar{x}_{jy}, \bar{c}_{ky}\right)}{M . \min_{\substack{j,k \\ j \neq k}}^{m} \left\| \bar{c}_{jy} - \bar{c}_{ky} \right\|^2} \qquad \text{...(B-16)}$$

Bio-Data

Name : **Mohammad Fazle Azeem**
Date of Birth : 1st, July, 1966

Academic Qualifications:

1. B.E. degree from Electrical Engineering Department, M.M.M. Engineering College, University of Gorakhpur, Gorakhpur, 1987
2. Masters degree in Instrumentation and Control, Electrical Engineering Department, ZHCOE & T., Aligarh Muslim University, Aligarh., 1991

Employment:

1. Teaching Assistantship, Computer Science Department, M.M.M. Engineering College, University of Gorakhpur, Gorakhpur, July'1987-July'1988.
2. Lecturer, Electrical Engineering Department, ZHCOE & T., Aligarh Muslim University, Aligarh, December'89-December'95.
3. Senior Lecture, Electrical Engineering Department, ZHCOE & T., Aligarh Muslim University, Aligarh, December'95-till date.

Research Publications during the Ph.D. Program

(a) Conference Papers

1. M.F. Azeem, M. Hanmandlu, N. Ahmad, "Evolutive Learning Algorithms for Fuzzy modeling".*ANNIE-2000, St.Louis, Missouri, USA, Nov 05-08, 2000*
2. M.F. Azeem, M. Hanmandlu, N. Ahmad, "Fuzzy Modeling of Fluidized Catalytic Cracking Unit", *Intl. Conf. on Cognitive System, New Delhi, India, Dec. 15-18, 1999*
3. M.F. Azeem, M. Hanmandlu, N. Ahmad, "Extension in Functional Equivalence of RBF Networks and Fuzzy Inference Systems", *2nd Intl. Conf. on Infor. Tech., Bhubaneswar, India, December 20-23*, 1999.
4. M. F. Azeem, M. Hanmandlu, N. Ahmad, "Modified Mountain Clustering in Dynamic Fuzzy Modeling", *2nd Intl. Conf. on Infor. Tech., Bhubaneswar, India, December 20-23*, 1999.

5. M. F. Azeem, M. Hanmandlu, N. Ahmad, "A New Criteria for Input Variable Identification of Dynamical Systems", *in proc. of IEEE Region 10 Intl. Conf., TENCON'98, New Delhi, India*, pp. 230-233.1998.

6. M. F. Azeem, M. Hanmandlu, N. Ahmad, "An Application of Fuzzy Clustering in Dynamic Fuzzy Modeling", *in proc. 3rd Intl. Conf. on Appl. of Fuzzy systems and Soft Computing ICAFS-98, Siegen, Germany*, October 5-7, pp. 143-155, 1998.

(b) Journals Papers (To appear/Communicated)

7. M.F. Azeem, M. Hanmandlu, N. Ahmad, "Generalization of Adaptive Neuro-Fuzzy Inference Systems", *IEEE Trans. on Neural Networks, vol 11, No. 6, November 2000* (to appear.)

8. M.F. Azeem, M. Hanmandlu, N. Ahmad, "Unification of CRI and TS-Models", *IEEE Trans. on Fuzzy Systems,* (Submitted on 20th September'2000.).

9. M.F. Azeem, M. Hanmandlu, N. Ahmad, "Structure Identification of Generalized Adaptive Neuro-Fuzzy Inference Systems", *IEEE Trans. on Fuzzy systems,* (Submitted on 20th September'2000.)

10. M.F. Azeem, M. Hanmandlu, N. Ahmad, "$\mathcal{O}(n)$ Criteria for Input Identification", *Automatica* , (Submitted on 20th September'2000.).

11. M.F. Azeem, M. Hanmandlu, N. Ahmad, "Evolutive Learning for Fuzzy modeling", *IEEE Trans. on Evolutive Computations,* (Submitted on 4th October'2000.).

(c) Papers to be communicated

12. M.F. Azeem, M. Hanmandlu, N. Ahmad, "Dynamic Fuzzy Modeling of Fluidized Catalytic Cracking Unit". A chapter in an edited book.